Fundamentals of Information Security Series

C836

Instructor: Arthur Moore

Event time: Every Thursday at 6:00PM MST

WESTERN GOVERNORS UNIVERSITY

ONLINE. ACCELERATED. AFFORDABLE. ACCREDITED.

Instructor intro

- Contact info: Arthur.Moore@wgu.edu
- Number: 1-877-435-7948 Ext. 1554
- Office Hours:
- Monday: 8AM-5PM
- Tuesday: 8AM-5PM
- Wednesday: 8AM-5PM
- Thursday: 8AM-12PM AND 7PM-10PM
- Sunday: 6PM-10PM

As a reminder

- This web series is only to enhance/supplement the course material not to replace it please follow the guidelines as posted below.
- ***It's Highly recommended that you Don't schedule the first assessment until all the materials are covered***.
- -take the preassessment until constantly passing
- -review lessons 1-14 readings (pay close attention to chapters 1 – 6.)
- -complete lessons 1-14 quizzes.
- -complete lessons 1-14 exercises (pay close attention to chapters 1 – 6)
- -watch the videos that are posted in the course tips
- -go over the PowerPoints that are posted in the course tips

Software Development Vulnerabilities

- A number of common software development vulnerabilities can lead to security issues in our applications. These issues are all well known as being problematic from a security perspective, and the reasons the development practices that lead to them should not be used are a frequent topic of discussion in both the information security and software engineering communities.
- The main categories of software development vulnerabilities include:
  - Buffer overflows
  - Race conditions
  - Input validation attacks
  - Authentication attacks
  - Authorization attacks
  - Cryptographic attacks,

Software Development Vulnerabilities

- **Buffer overflows**: also referred to as buffer overruns, occur when we do not properly account for the size of the data input into our applications. If we are taking data into an application, most programming languages will require that we specify the amount of data we expect to receive and set aside storage for that data.
- **Race Conditions**: Race conditions occur when multiple processes or multiple threads within a process control or share access to a particular resource, and the correct handling of that resource depends on the proper ordering or timing of transactions.
- **Input Validation Attacks:** If we are not careful to validate the input to our applications, we may find ourselves on the bad side of a number of issues, depending on the particular environment and language being used. A good example of an input validation problem is the format string attack.

Software Development Vulnerabilities

- **Authentication Attacks:** When we plan the authentication mechanisms our applications will use, taking care to use strong mechanisms will help to ensure that we can react in a reasonable manner in the face of attacks. There are a number of common factors across the various mechanisms we might choose that will help make them stronger.
- **Authorization Attacks:** Just as we discussed when we looked at authentication, placing authorization mechanisms on the client side is a bad idea as well. Any such process that is performed in a space where it might be subject to direct attack or manipulation by users is almost guaranteed to be a security issue at some point. We should instead authenticate against a remote server or on the hardware of the device, if we have a portable device, where we are considerably more in control.
- **Cryptographic Attacks:** We leave ourselves open to failure if we do not pay close enough attention to designing our security mechanisms while we implement cryptographic controls in our applications. Cryptography is easy to implement badly, and this can give us a false sense of security.

Web Security

- **Client-Side Attacks:** Client-side attacks take advantage of weaknesses in the software loaded on our clients, or those attacks that use social engineering to trick us into going along with the attack. There are a large number of such attacks, but we will focus specifically on some that use the Web as an attack vehicle.
  - **Cross-site scripting (XSS)**: is an attack carried out by placing code in the form of a scripting language into a Web page, or other media, that is interpreted by a client browser, including Adobe Flash animation and some types of video files. When another person views the Web page or media, he or she executes the code automatically, and the attack is carried out.
  - **Cross-site request forgery (XSRF)**: attack is similar to XSS, in a general sense. In this type of attack, the attacker places a link, or links, on a Web page in such a way that they will be automatically executed, in order to initiate a particular activity on another Web page or application where the user is currently authenticated. For instance, such a link might cause the browser to add items to our shopping cart on Amazon or transfer money from one bank account to another.
  - **Clickjacking**: is an attack that takes advantage of the graphical display capabilities of our browser to trick us into clicking on something we might not otherwise. Clickjacking attacks work by placing another layer over the page, or portions of the page, in order to obscure what we are actually clicking.

Web Security

- **Server-Side Attacks**: On the server side of the Web transaction, a number of vulnerabilities may cause us problems as well. Such threats and vulnerabilities can vary widely depending on our operating system, Web server software, various software versions, scripting languages, and many other factors. Across all of these, however, are several factors that are the cause of numerous security issues that are common across the various implementations we might encounter.
  - **Lack of input validation:** Structured Query Language (SQL) injection gives us a strong example of what might happen if we do not properly validate the input of our Web applications. SQL is the language we use to communicate with many of the common databases on the market today.
  - **Improper or inadequate permissions:** Particularly with Web applications and pages, there are often sensitive files and directories that will cause security issues if they are exposed to general users. One area that might cause us trouble is the exposure of configuration files.
  - **Extraneous files:** When we move a Web server from development into production, one of the tasks often missed in the process is that of cleaning up any files not directly related to running the site or application, or that might be artifacts of the development or build process.

Database Security

- As we discussed when we went over Web security issues, the vast majority of Web sites and applications in use today make use of databases in order to store the information they display and process. In some cases, such applications may hold very sensitive data, such as tax returns, medical data, or legal records; or they may contain only the contents of a discussion forum on knitting.
  - A number of issues can cause trouble in ensuring the security of our databases. The canonical list includes the following:
  - Unauthenticated flaws in network protocols
  - Authenticated flaws in network protocols
  - Flaws in authentication protocols
  - Unauthenticated access to functionality
  - Arbitrary code execution in intrinsic SQL elements
  - Arbitrary code execution in securable SQL elements
  - Privilege escalation via SQL injection
  - Local privilege escalation issues

Database Security

- **Protocol Issues**: We can look at the network protocols used to communicate with the database, some of which will need a set of credentials in order to use and some of which will not. In either case, there is often a steady stream of vulnerabilities for most any major database product and version we might care to examine. Such vulnerabilities often involve some of the more common software development issues, such as the buffer overflows we discussed at the beginning of this lesson.
- **Unauthenticated Access**: When we give a user or process the opportunity to interact with our database without supplying a set of credentials, we create the possibility for security issues.
- **Arbitrary Code Execution**: We can find a number of areas for security flaws in the languages we use to talk to databases. Generally, these are concentrated on SQL, as it is the most common database language in use. In the default SQL language, a number of built-in elements are possible security risks, some of which we can control access to and some of which we cannot.
- **Privilege Escalation**: In essence, privilege escalation is a category of attack in which we make use of any of a number of methods to increase the level of access above what we are authorized to have or have managed to gain on the system or application through attack. Generally speaking, privilege escalation is aimed at gaining administrative access to the software in order to carry out other attacks without needing to worry about not having the access required.