BSc (Hons) Computer Science

University of Portsmouth
Second Year

Ethical Hacking

M30239
Semester 2

Hugh Baldwin
hugh.baldwin@myport.ac.uk

Contents

Lecture - Introduction to Penetration Testing

09:00 22/01/24 Tobi Fajana

Weekly Teaching Materials

- 1 hour lecture
- 2 hour practical
- Instructional video (Guide for Labs)
- Compulsory Moodle quiz

Assessments

- Practical Exam, 2 hours (50%, 20/03/2024)
  - 2 Devices to exploit
  - 5 Vulnerabilities expected, give the name, software, risk rating, a brief description, and corrective actions for each exploit
- Muliple Choice Exam, 1 hour (50%, May/June)

CIA

- The three main properties which are protected by cyber security
- Confidentiality - Protecting information from being disclosed to unintended parties
- Integrity - Protecting information from being modified, intentionally or otherwise
- Availability - Ensuring the information is available to access when it is needed

Penetration Testing

- Black Box - No information about the target
- Grey Box - Some information about the target, but not all
- White Box - All information given, including source code, etc
- Timeframe - There is usually a fixed timeframe for the test
- Penetration Testing is similar to vulnerability assessment, but actual exploits the vulnerabilities

Port Scanning

- Scan every port on a server to check which ports are open
- Attempt to ping the server on each port
- If a response is given from a port, it must be open

Lecture - Information Gathering

09:00 29/01/24 Tobi Fajana

Active Information Gathering

- Directly interacting with the target
- You typically gather more information actively, but you are much more likely to get caught
- Active methods include
  - Probing the Network (Port scanning, service version enumeration)
  - Social Engineering (Password gathering with phishing)
  - Directory and Share scanning

Passive Information Gathering

- Avoid direct interaction as much as possible
- Much less likely to be caught, but less information is gathered
- Passive methods include
  - Using a Search Engine
  - Physical Observation (Looking over shoulder when typing a password)
  - DNS enumeration (Whois lookup, IP address lookup, shodan, etc)
  - OSINT Framework (Open Source Intellegence Network - Google Dorking, shodan, social media analysis)
  - GeoLocating people based upon images on social media
  - Searching on pastebin and other similar websites for the target
  - Looking on websites such as haveibeenpwnd.com to check if passwords for the target have been leaked

Lecture - Web Application Attacks

09:00 15/04/24 Tobi Fajana

Web Applications

A web server is either software or hardware whose sole purpose is to respond to HTTP requests with requested content, for example HTML web pages or JSON API content. Some web servers also support server-side scripting languages, such as PHP.

There are multiple options for web server software, but the most common two on the internet are Apache and Nginx. Both are still updated, and support most modern features, but by nature of being significantly older, Apache is much more prone to security issues. It is very important to know what server you are trying to exploit, as they are very different and have different vulnerabilities.

Attacks

Attacks can occur either on the client-side or server-side, which can cause different issues for the server's owner. For example, a client-side attack may steal credentials or other sensitive information from the client, without the server even knowing, but a server-side attack is usually more devastating since it would most likely have direct access to the database which the application uses to store all of it's information.

Finding Attack Vectors

There are several methods for finding attack vectors, namely

- Fuzzing
- Encoding
- Encryption
- etc

SQL Injection

One example of a server-side attack is an SQL injection attack, which works by tricking the database into executing unintended commands, or accessing unintended data using a normal query. The usual attack vector for an SQL injection is anywhere that user-input is used as part of a query, for example a search box or registration form.

Types of SQL Injection

There are three main types of SQL injection, namely

- In-Band SQL Injection - Data is returned through the same channel (page, input box, etc) that is used to inject the SQL code. This type of injection usually exploits issues with very simple queries, or by exploiting unions or errors in the way the query is written
- Inferential SQL Injection - Data is not returned at all through the web application, and so it is not possible to see the results of the attack'

- Out-of-Band SQL Injection - Data is returned through a different channel to the injection channel, for example through an email or notification

Another method of SQL injection is with a Union-based injection, which uses the `UNION` keyword to append another query onto the end of the intended query. This is quite easy to defend against, but can be very powerful if it is possible to exploit.

Time-blind SQL Injection

One method of exploiting a blind injection point is to use a time-delay or sleep function in the query, such that if the data you are expecting is in the database, the server will take longer to respond. For example, if you want to check if a user with the name 'administrator' exists in the database, we could use the following query `IF (SELECT user FROM users WHERE username='admin' AND SLEEP(10)) --`. In this case, if there is a user with the username 'administrator', the server will delay by 10 seconds before responding to the request.

Defending Against Injection

There are several methods to prevent injection attacks, for example

- Validate, filter and sanitize every input before executing
- Using parametrized queries to prevent escaping quotations
- Using server-side input validation
- Using permissions to limit the scope of each users access to the database
- Web Application Firewalls

Lecture - Binary Exploitation

09:00 22/04/24 Tobi Fajana

If you have permission to read the binary of a program, it is usually possible to extract some of the text which was compiled in. This is because `cat` will simply write out the binary to the terminal, and any strings would still be encoded as they were before the program was compiled. This means that any code that compares user input to a string could accidentally be insecure.

When you `cat` a binary, it will often also output the binary information added to the file by the compiler. This often includes information such as which compiler was used, what OS it was compiled by, etc.

While the program is running, it is also sometimes possible to view the stack and heap of the application, which will often include sensitive information, such as a string that is being compared against, or the result of some arithmetic being done in the background.

Buffer Overflow

With the techniques discussed previously, it may be possible to determine the length of a buffer used by the program to store user-input. This allows you to determine how long your input needs to be to overflow the buffer, which would then crash the program. On some platforms, this can also be used to overwrite the program code and cause unintended code to run in place of the program. This typically affects availability, but depending upon the memory layout of the program, it could possibly affect the integrity or confidentiality of some data.

Defending Against Buffer Overflow

There are several methods for protecting against buffer overflows, such as

- Address Space Layout Randomization (ASLR)
- Data Execution Protection
- Structured Exception Handling
- Patch devices/ software
- Validate Data

Lecture - Windows AD & Password Attacks

09:00 29/04/24 Tobi Fajana

Active Directory

Active Directory (AD) is a hierarchical structure which stores information about objects on the network. Each object could be a user, computer, printer, file share, group of users, etc. It provides centralised management for users and computers, which makes it easier for administrators to manage, and users to use as they can log in on any computer.

An Active Directory domain is made up of several services

- Domain Services
- Lightweight Domain Services (& Lightweight Directory Access Protocol)
- Certificate Services
- Active Directory Federation Services
- Rights Management Services

Security Identifiers

A security identifier (SID) is a unique value which is used to identify a so-called ‘security principal’ (e.g. security group) within Windows. The last 4 digits of an SID are the ‘Relative ID’, and the most important is `0500`, as this is the relative ID of an administrator account.

When attacking an AD network, the first thing you should do is *enumerate* all of the devices on the network, and gather their SIDs. This allows you to find the Domain Controller on the network, and if you can gain control of that, you have effective control over the entire AD network.

Domain Services

Active Directory Domain Services (ADDS) uses a hierarchical structure consisting of domains, trees and forests. A domain is a group of objects, such as users and computers, which share the same AD Database. A tree is a group of one or more domains, and a forest is a group of multiple trees. A forest consists of shared catalogues, schemas, application information and domain configurations. A schema defines an object’s class and attributes within a forest.

Domain Controller

The domain controller of an AD network is simply the server running ADDS. There can be many domain controllers on a network, but only one can be the primary controller at a time. This allows you to have complete fail-over if one of the servers were to fail. It also allows some basic load balancing in larger domains, but since one of the controllers is always primary, any secondary servers must be in constant communication with the primary.

Password Attacks

There are several methods of obtaining passwords, but the easiest will always be social engineering, to get people to willingly hand over their passwords. Second to that, you can brute-force or crack a password.

Password Guessing

While this is the simplest method, it is not fool-proof. This is because while it is a very simple method, it has many downsides–

- Easy to detect
- Takes a long time
- Uses lots of bandwidth
- Could be blocked at either end

Sniffing

It is possible to 'sniff' packets as they move across a network. This typically only works if the passwords are sent in plaintext, so not for any websites which use HTTPS, which is basically all of them at this point, since browsers now enforce secure connections. It also requires direct access to the network, which may be easy if the target is using unprotected wifi, but may be difficult if it is a physical network in a secure building.

Malware

If malware could be installed to the targets device, it is possible to use a *keylogger*, which records all of the keypresses made on a computer, which may include them inputting passwords.