MDSE-21 GROUP FOUR		
NAME	**REG No**	**STUDENT No**
FAHAD Guma	2021/HD05/2350U	2100702350
MUSA Rahim	2021/HD05/2354U	2100702354
FLORENCE Nanteza	2021/HD05/4137U	2100704137
MUGOYA Dihfahsih	2021/HD05/2353U	2100702353

An overview of our presentation

1. TCP Client - Server Architecture
2. Mail Client using python
3. Simple Web Server
4. Mpeg - Dash implementation
5. Penetration Test
6. The code of each section will be run as a demo to each part.

TCP Client - Server Architecture

TCP Client - Server Architecture

The client/server architecture typically consists of a collection of computers connected by a communication network. The functions of the information system are performed by processes (computer programs) that run on these computers and communicate through the network.

We shall use **sockets** for this

Network applications use sockets to communicate over a TCP/IP network. A socket is an abstraction that represents an endpoint of a connection. Because a socket is bidirectional, data can be sent as well as received through it.

A socket has three attributes:

- The network address (the IP address) of the system
- The port number identifying the process (a process is a computer program running on a computer) that exchanges data through the socket
- The type of socket (such as stream or datagram) identifying the protocol for data exchange

Email Client using python

Email Client Explained

- Email client is a desktop application that enables configuring one or more email addresses to receive, read, compose and send emails from that email address(s) through the desktop interface.
- It provides a central interface for receiving, composing and sending emails of configured email address(s).
- In short, an email client is a computer program used to read and send electronic messages.
- Email client is also known as email reader or mail user agent (MUA).

What Can I Do With an Email Client?

- The email client lets you read, organize, and reply to messages as well as send new emails.
- To organize email, email clients typically offer folders, labels, or both.
- An integrated search engine lets you find messages by details such as senders, subjects, times of receipt, and content.
- In addition to email text, email clients also handle attachments, so you can send and receive computer files (such as images, documents or spreadsheets) via email.

Email Client and Email Servers

- Email clients can use a number of protocols to send and receive emails via email servers.
- To send an email, email clients use SMTP (Simple Mail Transfer Protocol) almost exclusively.
- The messages are either stored locally on your computer (typically when POP, or Post Office Protocol) is used to download mail from the server), or emails and folders are synchronized with the server (usually when the IMAP and Exchange protocols are employed).
- With IMAP (Internet Message Access Protocol) and Exchange, email clients accessing the same account see the same messages and folders, and all actions automatically synchronize.

Common Email Clients

- Popular email clients
  - Microsoft Outlook
  - Mozilla Thunderbird
  - macOS Mail
  - IncrediMail
  - Mailbox
  - iOS Mail
- The most popular web-based email client is Gmail; others include Yahoo! Mail and Outlook.com

Simple Mail Transfer Protocol(SMTP)

- Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.
- Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

Four basic steps for sending emails using Python

- Set up the SMTP server and log into your account.
- Create the MIMEMultipart message object and load it with appropriate headers for From, To and Subject fields
- Add your message body
- Send the message using SMTP server object

Simple Web Server

Web server in python

In the infrastructure of the internet, the server is one part of the client-server model. When a client browser visits a web page, it makes an HTTP request to the server containing the files needed to operate a website.

The server listens to the client’s request, processes it, and responds with the required files to present the web page. This content could be HTML (the text and media you see on a website) and JSON (applications).

404 “page-not-found”

You might have encountered a few server error codes in your time browsing the internet - “file not found” or 404 being a more popular one. In these cases, the server has trouble accessing certain files. With a 404 error, the particular file is missing.

There are more nuances to web servers, including classification into static and dynamic web servers.

For example, static web servers only return files as they are, with no extra processing. Dynamic web servers introduce databases and application servers, which you can proceed to once you've got the hang of static servers.

How Do You Create a Simple Python Web Server?

Launching a Python web server is quick and straightforward, and it’ll only take a few minutes for you to get up and to run. All it takes is one line of code to get the simplest of local servers running on your computer.

By local testing, your system becomes the server to the client that is your browser, and the files are stored locally on your system. The module you'll be using to create a web server is Python's http server. There is one caveat to this: it can only be used as a static file server. You'll need a Python web framework, like Django, to run dynamic web servers.

Creating a Custom Web Server Using Python

A custom web server allows you to do more than a built-in web server. The code you're about to see will show you a lot about some important functions and processes.

MPEG - DASH

The applicability of DASH in web services

What is MPEG-DASH

MPEG - **M**oving **P**icture **E**xpert **G**roup

DASH - **D**ynamic **A**daptive **S**treaming over **H**TTP

MPEG-DASH is one of the most popular video-streaming protocols and is widely used to deliver media either via Video on Demand (VOD) or Live Streaming and to various end-user devices, including smartphones, tablets, SmartTVs, gaming consoles, and more.

MPEG-DASH is an HTTP-based streaming protocol for delivering video to the end-user from an HTTP server. In MPEG-DASH, a video is split into segments and this information is recorded in an MPD.

The MPD is first delivered to the player, which uses it to request segments of the appropriate bitrate & resolution based on the network conditions and buffer fullness.

❖ YouTube MPEG-DASH

- Adaptive HTTP Streaming using 8 video resolution file in the server

Resolution	Pixels (width x height)
144p	256 x 144
240p	426 x 240
360p	640 x 360
480p	854 x 480
720p	1280 x 720
1080p	1920 x 1080
1440p	2560 x 1440
2160p	3840 x 2160

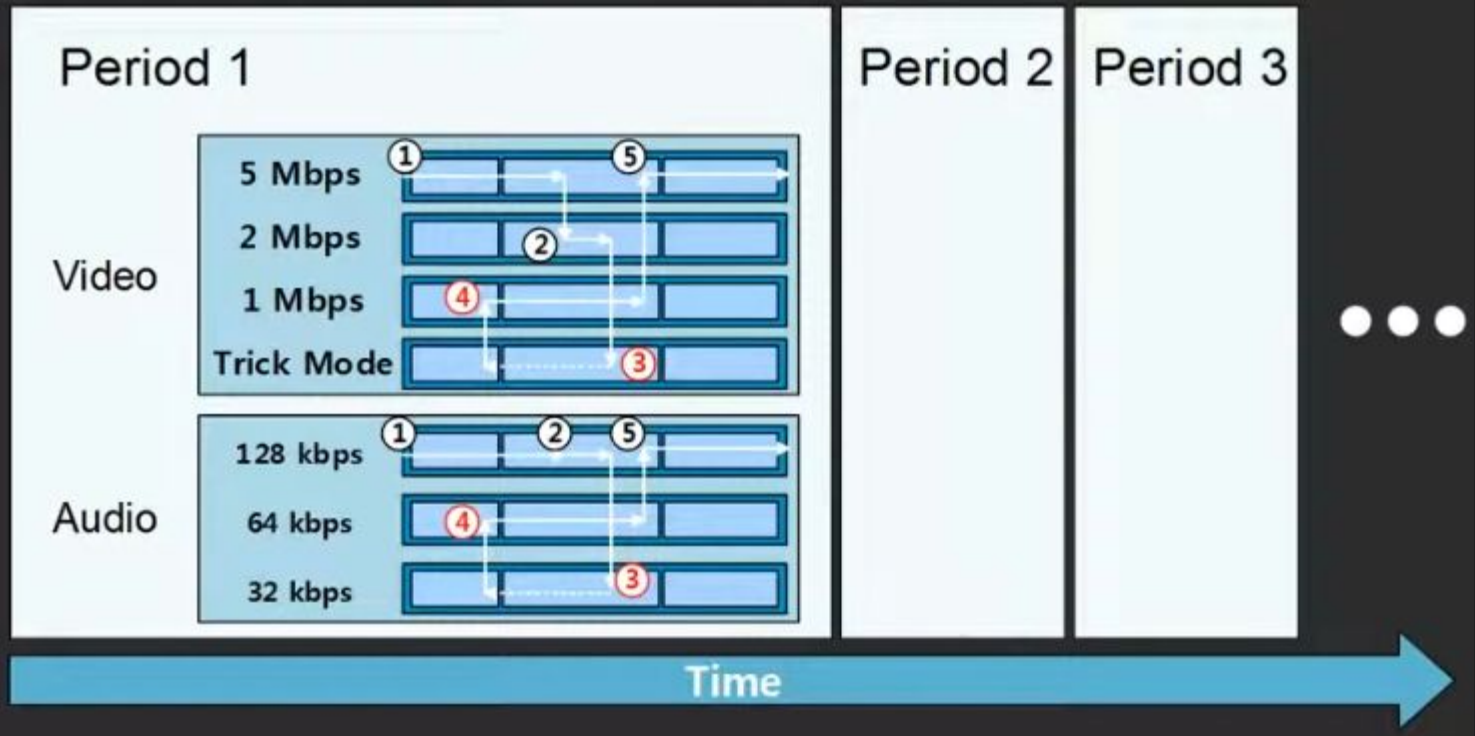
❖ Types of HTTP Video Streaming

- Regular HTTP Streaming: Completely downloads the video to the client device
- Progressive Download: Progressive downloading of the video at a fixed quality
- Adaptive Streaming over HTTP: Combination of Adaptive Video Quality (bitrate) Control & Progressive Downloading

❖ Progressive Downloading Advantages

- Mobile device may not be able to store the entire video due to limited memory space
- Savings in both Personal Data Usages and Internet Bandwidth
  - Many users to not watch the entire video (many watch less than 20% of the entire video)

❖ Progressive Downloading Advantages



Example of DASH operations
Numbered circles represent the action points controlled by the Client device

❖ MPEG-DASH specifications ISO/IEC 23009-1

- MPEG-DASH standard only defines MPD and the segment formats
  - MPD: Media Presentation Description
- MPD delivery, format of media-encoding that include the video segments, adaptive downloading, video playing control are determined by the application on the Client device

❖ MPEG-DASH specifications ISO/IEC 23009-1

- HTTP Server
  - MPD (Media Presentation Description)
  - Video and Audio data segments

❖ Playing on the DASH Client

1. DASH client requests for MPD from the Server

2. MPD is delivered to the Client using HTTP, email, thumb drive, broadcast, etc.

3. Client parses the MPD

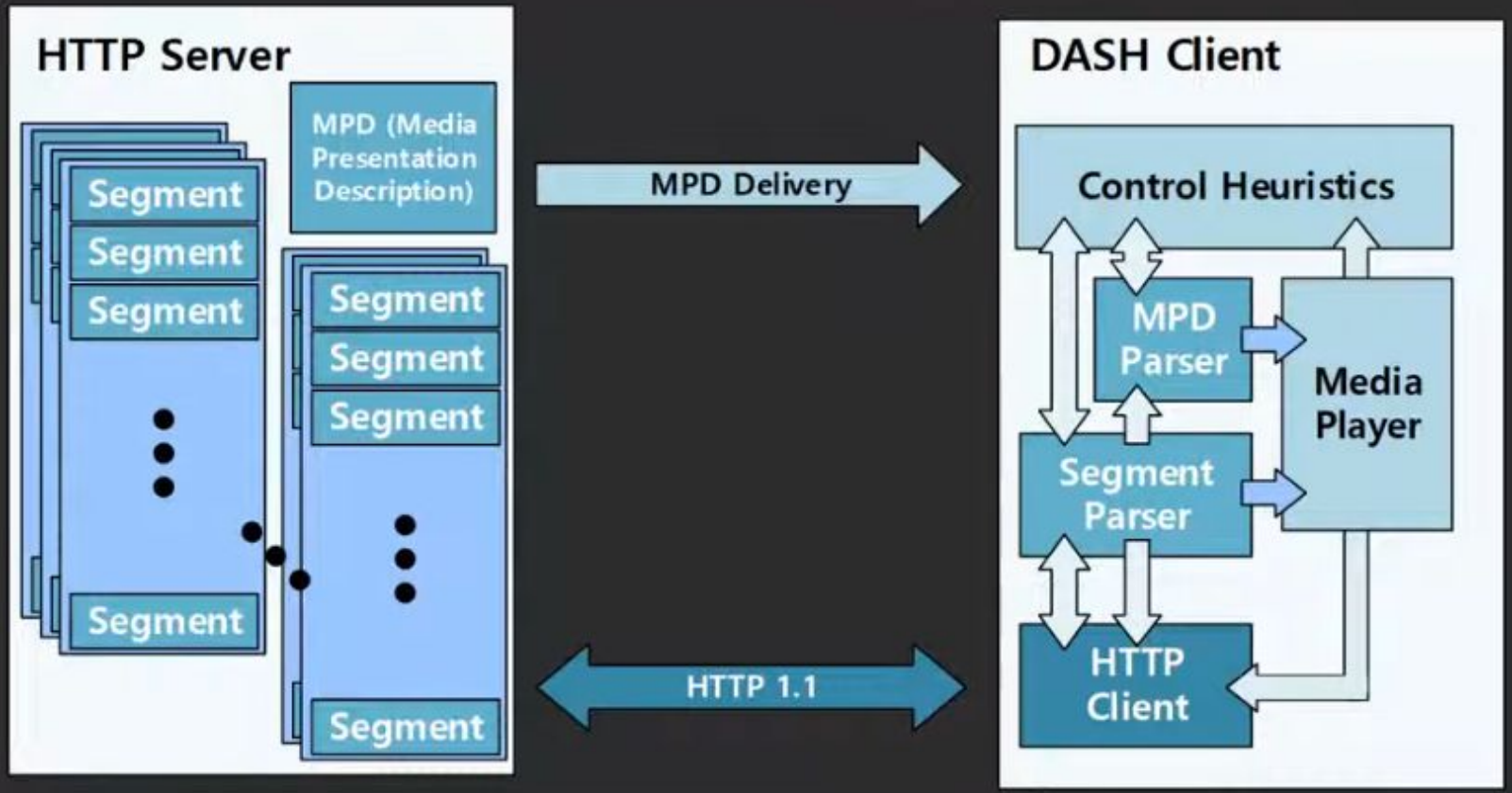
❖ Playing on the DASH Client

4. Client's MPEG-DASH application reads info
   - Program timing, media-content availability, media types, resolutions, minimum and maximum bandwidths, available multimedia resolution types, etc.
5. Client selects the appropriate encoded alternative and starts streaming the content by fetching the segments using HTTP GET requests

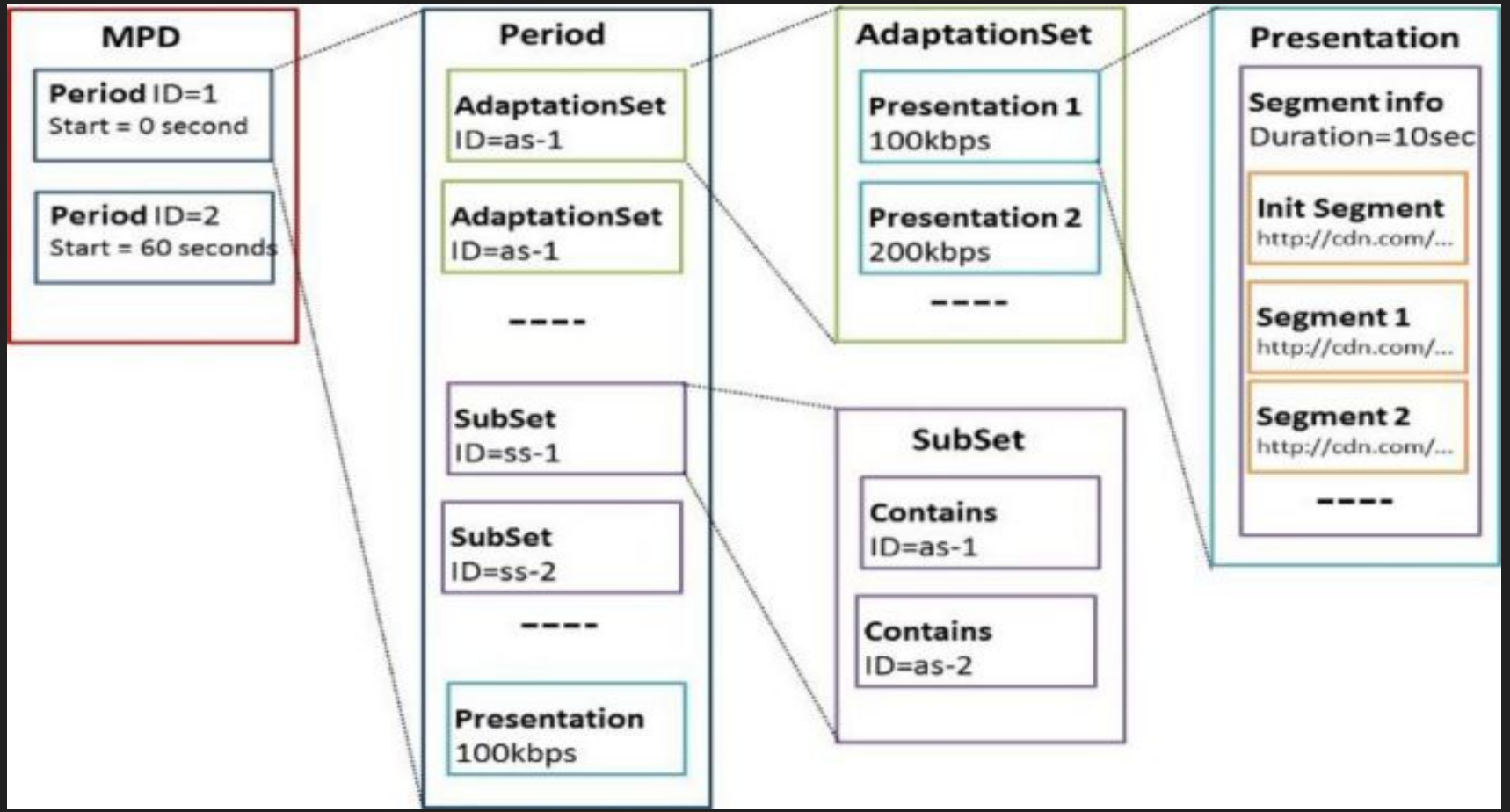
❖ Playing on the DASH Client

6. Appropriate buffering is used to compensate for network throughput variations
7. Client continues to fetch subsequent segments and also monitors the network bandwidth fluctuations
8. Client adapts to the available bandwidth by fetching segments with lower or higher bitrates to maintain an adequate buffer

❖ Playing on the DASH Client



MPEG-DASH system blocks and functionalities


MPD
Period ID=1
Start = 0 second
Period ID=2
Start = 60 seconds
Period
AdaptationSet
ID=as-1
AdaptationSet
ID=as-1

SubSet
ID=ss-1
SubSet
ID=ss-2

Presentation
100kbps
AdaptationSet
Presentation 1
100kbps
Presentation 2
200kbps

SubSet
Contains
ID=as-1
Contains
ID=as-2
Presentation
Segment info
Duration=10sec
Init Segment
http://cdn.com/...
Segment 1
http://cdn.com/...
Segment 2
http://cdn.com/...

❖ MPD Decoding & Playing Method

1. MPD contains multiple Periods

2. Period contains multiple Adaptation sets
   - Period is a program time interval which includes information on starting time and duration

❖ MPD Decoding & Playing Method

3. Adaptation set contains multiple Representations
   A. Representations have different bitrate encodings of video or audio of the same multimedia content
      - bitrate is based on resolution, number of channels, etc.
   B. Representation for Trick Mode can be included

❖ MPD Decoding & Playing Method

4. Representations consists of multiple media Segments information
   A. Segments are chunks of the media stream in a time order sequence
   B. Each segment has a URI (Uniform Resource Identifier)
   C. URI is used as the Server's address location from where the multimedia content can be downloaded from
   D. Download uses HTTP GET (may include byte ranges)

❖ MPD Decoding & Playing Method

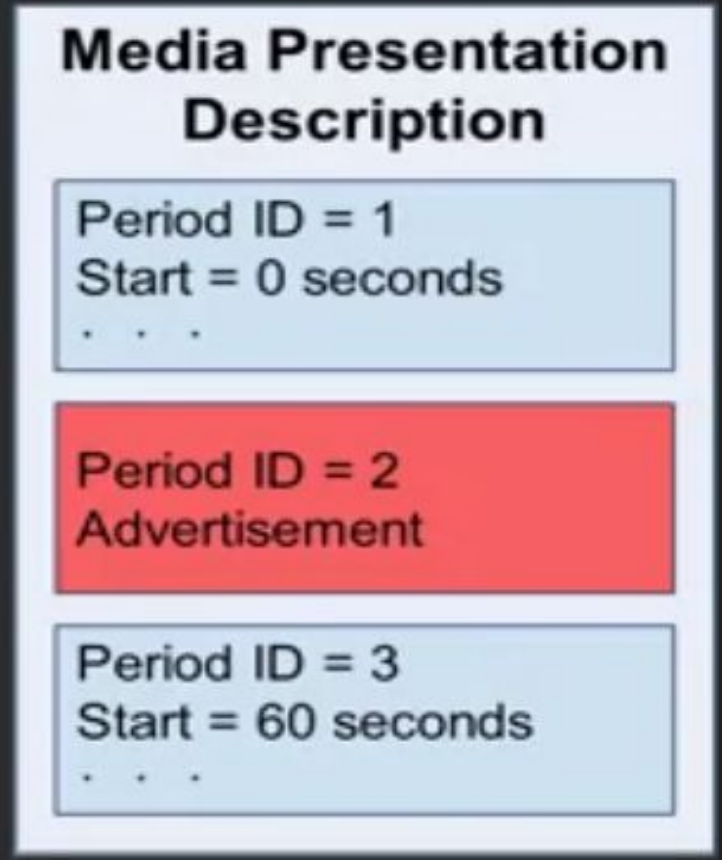
5. Segment information includes its initialization segment and information of multiple media segments

❖ MPEG-DASH Features

1. Switching and selectable streams
    - MPD provides adequate information to the client for selecting and switching between streams
        - Selecting one audio stream from different languages
        - Selecting video between different camera angles
        - Selecting the subtitles from provided languages
        - Dynamically switching between different bitrates of the same video camera

❖ MPEG-DASH Features

2. Ad insertion
   - Advertisements can be inserted as a period between periods or segment between segments in both on-demand and live cases



❖ MPEG-DASH Features

3. Compact manifest
   - Segments' address URLs can be signaled using a template scheme
   - Results in a compact MPD
4. Fragmented manifest
   - MPD can be divided into multiple parts or some of its elements can be externally referenced
   - Enables downloading MPD in multiple steps

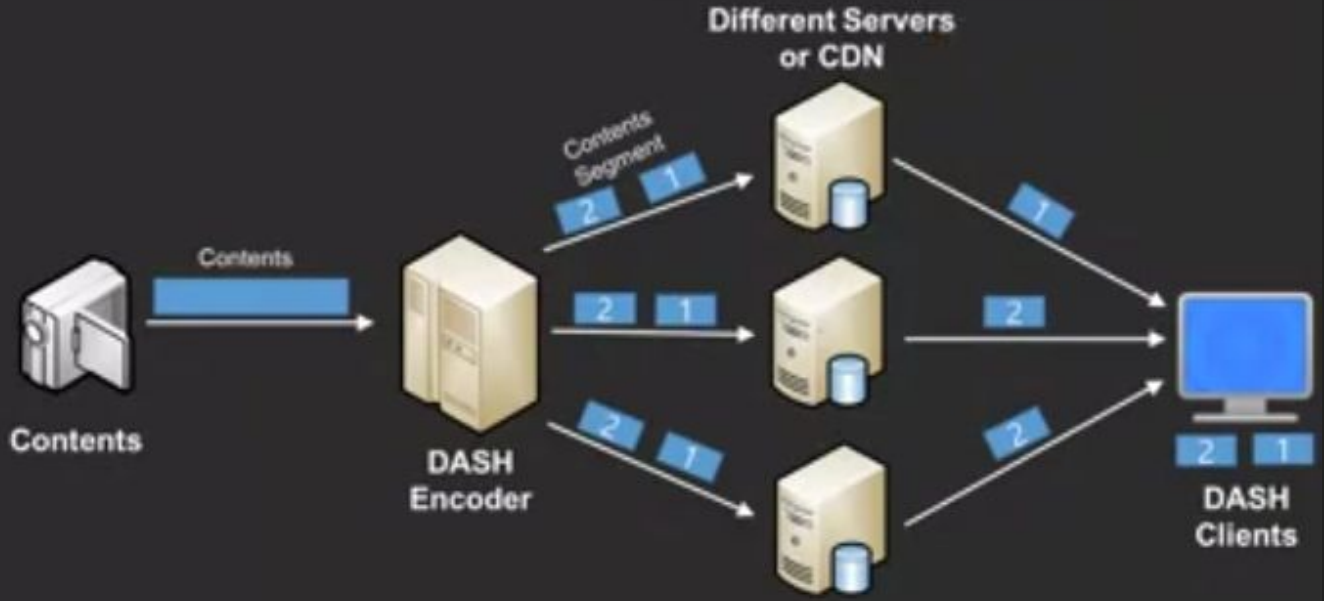
❖ MPEG-DASH Features

5. Segments with variable durations
   - Duration of segments can be varied
   - Duration information in Live Streaming
     - Duration of the Next segment can be provided in advance along with the delivery of the Current segment

❖ MPEG-DASH Features

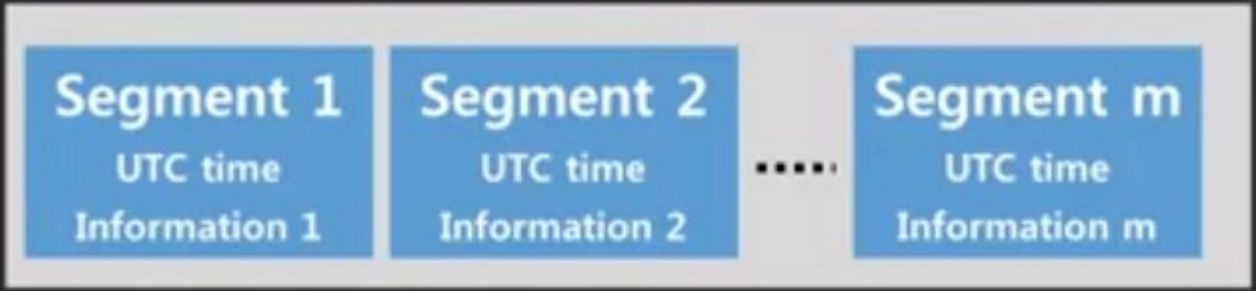
6. Multiple Base URLs

- Same content can be available at multiple URLs
- Client can stream from any URL



❖ MPEG-DASH Features

7. Clock-drift control for live sessions
   - UTC time can be included with each segment
   - UTC time enables the client to control its clock drift
     - UTC: Coordinated Universal Time



❖ MPEG-DASH Features

8. SVC (Scalable Video Coding) and MVC (Multiview Video Coding) support
   - MPD provides adequate information regarding the decoding dependencies between representations
   - This info can be used for streaming any (e.g., SVC, MVC) multilayer coded streams

❖ MPEG-DASH Features

9. A flexible set of descriptors
    - Descriptors describe the content rating, components' roles, accessibility features, camera views, frame packing, audio channels' configuration
10. Subsetting adaptation sets into groups
    - Grouping occurs according to the content author's guidance

❖ MPEG-DASH Features

11. Quality metrics for reporting the session experience
    - MPEG-DASH standards has a set of quality metrics
    - Metrics are used by the client to measure and report back to a server
12. Most of these features are provided in flexible and extensible ways
    - Enables MPEG-DASH to be used in flexible ways in various future apps

References

1. A. C. Begen, T. Akgul, and M. Baugher, "Watching Video over the Web, Part 1: Streaming Protocols," IEEE Internet Computing, Mar./Apr. 2011.

2. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," IEEE MultiMedia, Oct.Dec. 2011.

3. J. Anorga, S. Arrizabalaga, B. Sedano, M. L. Alonso-Arce, and J. Mendizabal, "YouTube's DASH implementation analysis," in Proc. IEEE CCSC, 2015.

4. D. K. Krishnappa, D. Bhat, and M. Zink, "DASHing YouTube an analysis of using DASH in YouTube video service," in Proc. 38th IEEE Conf. LCN, 2013.

5. Youtube, http://www.youtube.com Wikipedia, http://www.wikipedia.org

6. Image Sources Monitor icon, By No machine-readable author provided. Mobius assumed (based on copyright claims). [Public domain], via Wikimedia Commons

Penetration Testing

Select a network environment of your choice and perform a penetration test and write a report.

What is penetration testing

- A penetration test is a set of authorized cyber attacks, in order to discover and verify vulnerability of an information system.
- Penetration Testing or Ethical Hacking simulates a hacker targeting specific resources
- Different types of testing depending on the amount of knowledge about the target(s)
- It can involve the attempted breaching of any number of system applications to uncover vulnerabilities, such as unsanitized inputs that are susceptible to code injection attacks.

The need for pentesting

- Pentesting identifies the threats that might expose the confidentiality of an organization.
- Expert pentesting provides assurance to the organization with a complete and detailed assessment of organizational security.
- Pentesting assesses the network's efficiency by producing huge amount of traffic and scrutinizes the security of devices such as firewalls, routers, and switches.
- Changing or upgrading the existing infrastructure of software, hardware, or network design might lead to vulnerabilities that can be detected by pentesting.
- Potential threats are increasing significantly; pentesting is a proactive exercise to minimize the chance of being exploited.
- Pentesting ensures whether suitable security policies are being followed or not.

How it is performed

- Determine the scope
  - Web application pentest
  - End user and social engineering attacks
  - DDOS and performance tests
  - Network infrastructure tests
  - External and Internal network tests
  - Mobile application tests
  - Visualization system pentests
  - Database pentest

How it is performed

- Perform the test (stages)
  - Information gathering
  - Threat-modeling
  - Vulnerability analysis
  - Exploitation
  - Post Exploitation phase
  - Reporting phase

Network Penetration testing

- Network penetration testing is a security service that identifies security vulnerabilities in networks, systems, hosts, and devices by purposefully using malicious techniques to test the network’s security responses.
- The objective of network penetration testing is to identify security exploits that put an organization at risk of a data breach before hackers can discover and exploit them

Information gathering

During information gathering, the following tools were used to gain more information about the target organization network infrastructure and network resources being used.

- **Whois Lookup**
  - We used whois to identify the location of the servers, IP address, Server Type, contact information that is exposed to the public etc.
- **Wappalyzer**
  - We used wappalyzer to get information about the technologies used to develop the apps used by the organization.
- **Robtex**
  - We used it to gain DNS information about the target network applications

Information gathering

- **Subdomain Enumeration**
  - We developed a tool using python to do the subdomain enumeration using dns.resolver
- **Discovering Sensitive Files**
  - To discover sensitive information we developed a tool in python using urllib with the help of task threading so we can run multiple tasks at ago.
- **Port Scanning**
  - Using sockets and queuing , we developed a tool in python to scan all available open ports on a given IP address.

For successful penetration testing, the above tools and resources helped us in expanding the horizon of our successful test and to provide exposure to hidden targets that may be useful while assessing.

Threat Modelling

This is the third phase of PTES. Threat modeling approach is required for correct execution of penetration testing. Threat modeling can be used as part of a penetration test or it may be a separate engagement.

This helps in itemizing potential threats that an organization may face based on a number of factors. In case we are using threat modeling as part of penetration test, then the information gathered in the second phase would be rolled back into the first phase.

Vulnerability Analysis

A Vulnerability analysis is the systematic evaluation of potential and existing threats and flaws in the organization systems, networks, applications, hardwares, and other parts of the IT ecosystem.

Vulnerability analysis is one of the most effective technique for identifying security holes in the target organization. During vulnerability analysis, the following steps were followed;

- Vulnerability Identification which we had acquired from the previous phase.
- Assessing the nature and potential of a successful exploitation.
-

Exploitation

While conducting the exploitation, the following activities were performed

- Obtained historical breached account credentials to leverage against all company login pages.
- Attempted a “credential stuffing” attack against Google Suit accounts, which was unsuccessful. However, we gained as to a list of username.
- Tried password cracking using a tool we developed in python and that went through breach compilation to get posible usernames and passwords.
- Tried out a DDOS attack on one of the test servers which was successful

Post-Exploitation Phase

After exploitation, we did a clean up in the following ways;

- Removed all executable, scripts and temporary file from the compromised system.
- Returned to original values and system settings and application configuration parameters if they were modification was done during the assessment.
- Removed all backdoors created
- Removed any user accounts created for connecting to compromise systems.

Reporting

After doing a success penetrating we came up with a report detailing the following key points.

- The seriousness of the risks emanating from the vulnerabilities discovered
- The tools that can successfully penetrate the system.
- Highlighting those points where security had been implemented correctly
- vulnerabilities that need to be corrected and how to prevent future attacks