Nama:hilman nurharish
Nim:1310651028
Kelas:B

Software Development Security

INTRODUCTION

Software is everywhere: not only in our computers but also in our houses, our cars, and our medical devices, and all software programmers make mistakes.
As software has grown in complexity, the number of mistakes has grown along with it.
Developing software that is robust and secure is critical: this chapter will show how to do that. We will cover programming fundamentals such as compiled versus interpreted languages, as well as procedural and Object-Oriented Programming languages. We will discuss application development models such as the Waterfall Model, Spiral Model, and Extreme Programming (XP) and others. We will describe common software vulnerabilities, ways to test for them, and maturity frameworks to assess the maturity of the programming process and provide ways to improve it.

Compilers, interpreters, and bytecode

Compilers take source code, such as C or Basic, and compile it into machine code. Interpreted languages differ from compiled languages: interpreted code is compiled on the fly each time the program is run. Bytecode, such as Java bytecode,
is also interpreted code. Bytecode exists as an intermediary form (converted from source code) but still must be converted into machine code before it may run on the CPU.

Types of publicly released software

Once programmed, publicly released software may come in different forms (such as
with or without the accompanying source code) and released under a variety of licenses.

Open and closed source software

Closed source software is software typically released in executable form: the source
code is kept confidential. Open source software publishes source code publicly. Proprietary
software is software that is subject to intellectual property protections such as patents or copyrights.

Free Software, Shareware, and Crippleware

Freeware is a software, which is free of charge to use. Shareware is a fully functional proprietary software that may be initially used free of charge. If the user continues to use the Shareware for a specific period of time specified by the license (such as 30
days), the Shareware license typically requires payment. Crippleware is a partially functioning proprietary software, often with key features disabled. The user is typically required to make a payment to unlock the full functionality.

APPLICATION DEVELOPMENT METHODS

As software has grown in complexity, software programming has increasingly become a team effort. Team-based projects require project management: providing a project framework with deliverables and milestones, divvying up tasks, team communication,
progress evaluation and reporting, and (hopefully) a final delivered
product.

Waterfall Model

The Waterfall Model is a linear application development model that uses rigid phases; when one phase ends, the next begins. Steps occur in sequence, and the unmodified Waterfall Model does not allow developers to go back to previous steps.
It is called the waterfall because it simulates water falling: it cannot go back up.
A modified Waterfall Model allows a return to a previous phase for verification or validation, ideally confined to connecting steps.

Spiral

The Spiral Model is a software development model designed to control risk. The Spiral
Model repeats steps of a project, starting with modest goals and expanding outward in ever-wider spirals (called rounds). Each round of the spiral constitutes a project, and each round may follow traditional software development methodology such as modified waterfall. A risk analysis is performed each round. Fundamental flaws in the project or process are more likely to be discovered in the earlier phases,
resulting in simpler fixes. This lowers the overall risk of the project: large risks should be identified and mitigated.

Agile Software Development

Agile Software Development evolved as a reaction to rigid software development
models such as the Waterfall Model. Agile methods include Extreme Programming
(XP). Agile embodies many modern development concepts, including more flexibility,
fast turnaround with smaller milestones, strong communication within the team,
and more customer involvement.

Extreme Programming

Extreme Programming (XP) is an Agile development method that uses pairs of programmers
who work off a detailed specification. There is a high level of customer
involvement and constant communication.

Rapid Application Development

Rapid Application Development (RAD) rapidly develops software via the use of prototypes,
"dummy" GUIs, back-end databases, and more. The goal of RAD is quickly
meeting the business need of the system; technical concerns are secondary. The customer
is heavily involved in the process.

SDLC

The Systems Development Life Cycle (SDLC, also called the software development
life cycle or simply the system life cycle) is a system development model. SDLC is
used across the industry, but SDLC focuses on security when used in context of the
exam. Think of "our" SDLC as the "secure systems development life cycle": the
security is implied.

FAST FACTS

The following overview is summarized from NIST SP 800-14:

• Prepare a security plan: Ensure that security is considered during all phases of the IT system
life cycle and that security activities are accomplished during each of the phases.
• Initiation: The need for a system is expressed and the purpose of the system is documented.
• Conduct a sensitivity assessment: Look at the security sensitivity of the system and the

information to be processed.
• Development/acquisition: The system is designed, purchased, programmed, or developed.
• Determine security requirements: Determine technical features (like access controls),
assurances (like background checks for system developers), or operational practices
(like awareness and training).
• Incorporate security requirements into specifications: Ensure that the previously
gathered information is incorporated in the project plan.
• Obtain the system and related security activities: May include developing the system's
security features, monitoring the development process itself for security problems,
responding to changes, and monitoring threats.
• Implementation: The system is tested and installed.
• Install/turn-on controls: A system often comes with security features disabled. These
need to be enabled and configured.
• Security testing: Used to certify a system and may include testing security
management, physical facilities, personnel, procedures, the use of commercial or inhouse
services (such as networking services), and contingency planning.
• Accreditation: The formal authorization by the accrediting (management) official for
system operation and an explicit acceptance of risk.
• Operation/maintenance: The system is modified by the addition of hardware and software
and by other events.
• Security operations and administration: Examples include backups, training,
managing cryptographic keys, user administration, and patching.
• Operational assurance: Examines whether a system is operated according to its current
security requirements.
• Audits andmonitoring: A systemaudit is a one-time or periodic event to evaluate security.
Monitoring refers to an ongoing activity that examines either the system or the users.

• Disposal: The secure decommission of a system.
• Information: Information may be moved to another system, archived, discarded, or
destroyed.
• Media sanitization: There are three general methods of purging media: overwriting,
degaussing (for magnetic media only), and destruction.1

OBJECT-ORIENTED PROGRAMMING

Object-Oriented Programming (OOP) uses an object metaphor to design and write computer programs. An object is a "black box" that is able to perform functions and sends and receives messages. Objects contain data and methods (the functions they perform). The object provides encapsulation (also called data hiding): we do
not know, from the outside, how the object performs its function. This provides security
benefits: users should not be exposed to unnecessary details.

Cornerstone Object-Oriented Programming concepts

Cornerstone Object-Oriented Programming concepts include objects, methods, messages,
inheritance, delegation, polymorphism, and polyinstantiation. We will use an example object called "Addy" to illustrate the cornerstone concepts. Addy is an object that adds two integers; it is an extremely simple object, but has enough complexity
to explain core OOP concepts. Addy inherits an understanding of numbers and math from his parent class (the class is called mathematical operators). One specific
object is called an instance. Note that objects may inherit from other objects, in addition to classes.

In our case, the programmer simply needs to program Addy to support the method of addition (inheritance takes care of everything else Addy must know). Figure 4.1 shows Addy adding two numbers.

"1þ2" is the input message; "3" is the output message. Addy also supports delegation:
if he does not know how to perform a requested function, he can delegate that request to another object (called "Subby" in Figure 4.2).

Addy also supports polymorphism (based on the Greek roots "poly" and "morph," meaning many and forms, respectively): he has the ability to overload his plus (þ) operator, performing different methods depending on the context of

the input message. For example, Addy adds when the input message contains
“numberþnumber”;
polymorphism allows Addy to concatenate two strings when the
input message contains “stringþstring,” as shown in Figure 4.3.

Software vulnerabilities

Programmers make mistakes: this has been true since the advent of computer
programming.
The number of average defects per line of software code can often be reduced,
though not eliminated, by implementing mature software development practices.

Types of software vulnerabilities

This section will briefly describe common application vulnerabilities. An
additional
source of up-to-date vulnerabilities can be found at “2011 CWE/SANS Top 25
Most
Dangerous Programming Errors,” available at http://cwe.mitre.org/top25/; the
following
summary is based on this list. CWE refers to Common Weakness Enumeration,
a dictionary of software vulnerabilities by MITRE (see http://cwe.mitre.org/).
SANS is the SANS Institute; see http://www.sans.org.

• Hard-coded credentials: Backdoor username/passwords left by programmers in
production code
• Buffer overflow: Occurs when a programmer does not perform variable bounds
checking
• SQL injection: Manipulation of a back-end SQL server via a front-end Web
server
• Directory path traversal: Escaping from the root of a Web server (such as/var/
www) into the regular file system by referencing directories such as “../..”
• PHP Remote File Inclusion (RFI): Altering normal PHP URLs and variables
such
as “http://good.example.com?file¼readme.txt” to include and execute remote
content, such as http://good.example.com?file¼http://evil.example.com/bad.php
• Cross-Site Scripting (XSS): Third-party injection of a script into a Web page
within the security context of a trusted site

• Cross-Site Request Forgery (CSRF or sometimes XSRF): Third-party submission
of predictable content to a Web application within the security context of an
authenticated user[3]

Cross-Site Scripting and Cross-Site Request Forgery

Cross-Site Scripting and Cross-Site Request Forgery are often confused. They are
both Web attacks: the difference is XSS executes a script in a trusted context:

```
<script>alert(“XSS Test!”);</script>
```

The previous code would pop up a harmless “XSS Test!” alert. A real attack would include more JavaScript, often stealing cookies or authentication credentials.

Disclosure

Disclosure describes the actions taken by a security researcher after discoveringa
software vulnerability. Full disclosure is the controversial practice of releasing vulnerability details publicly. Responsible disclosure is the practice of privately sharing vulnerability information with a vendor and withholding public release until a patch is available. Other options exist between full and responsible disclosure.

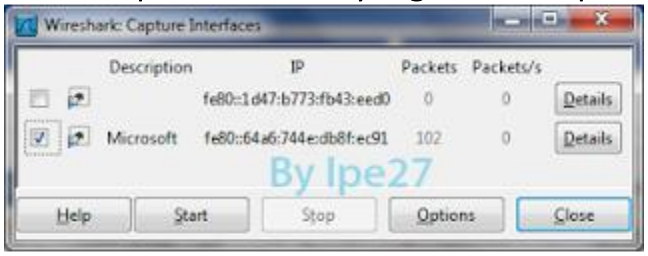
Software Capability Maturity Model

The Software Capability Maturity Model (CMM) is a maturity framework for evaluating
and improving the software development process. Carnegie Mellon University’s (CMU) Software Engineering Institute (SEI) developed the model. The goal of CMMis to develop a methodical framework for creating quality software that allows
measurable and repeatable results.

NO.2

Protokol HTTP

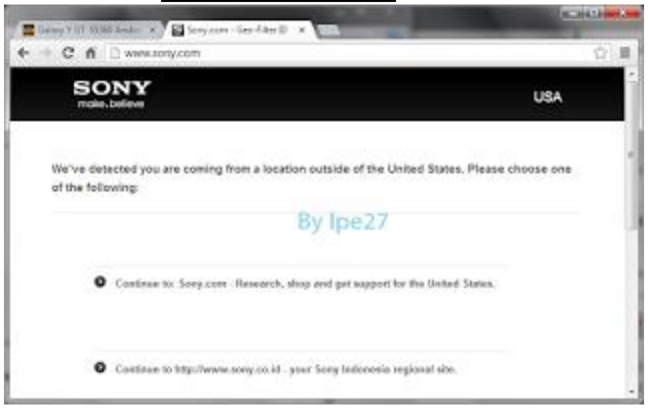
1. Buka web browser favorit anda, dapat menggunakan chrome, mozila, atau safari dan lain-lain.
2. Buka wireshark kemudian pilih interface yang akan dicapture, klik start.



Wireshark: Capture Interfaces

	Description	IP	Packets	Packets/s	
☐		fe80::1d47:b773:fb43:eed0	0	0	Details
☑	Microsoft	fe80::64a6:744e:db8f:ec91	102	0	Details

3. Pada browser bukalah situs www.sony.com.

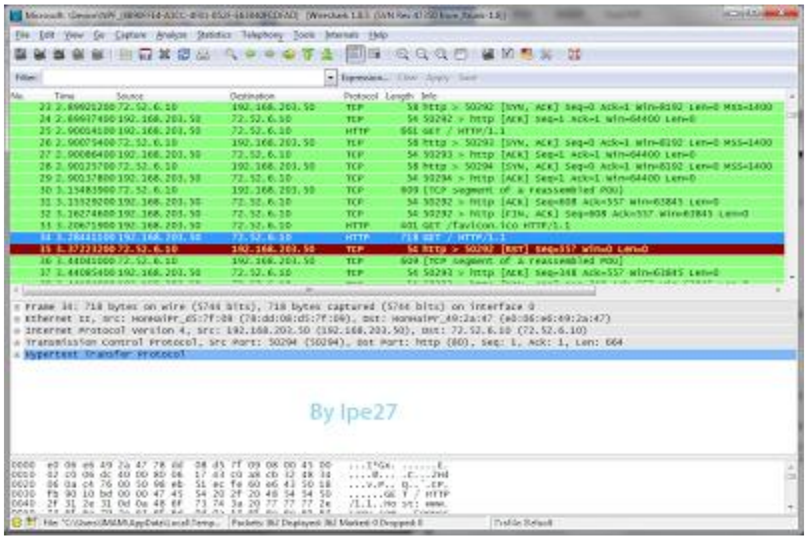


We've detected you are coming from a location outside of the United States. Please choose one of the following:

● Continue to: Sony.com - Research, shop and get support for the United States.

● Continue to http://www.sony.co.id - your Sony Indonesia regional site.

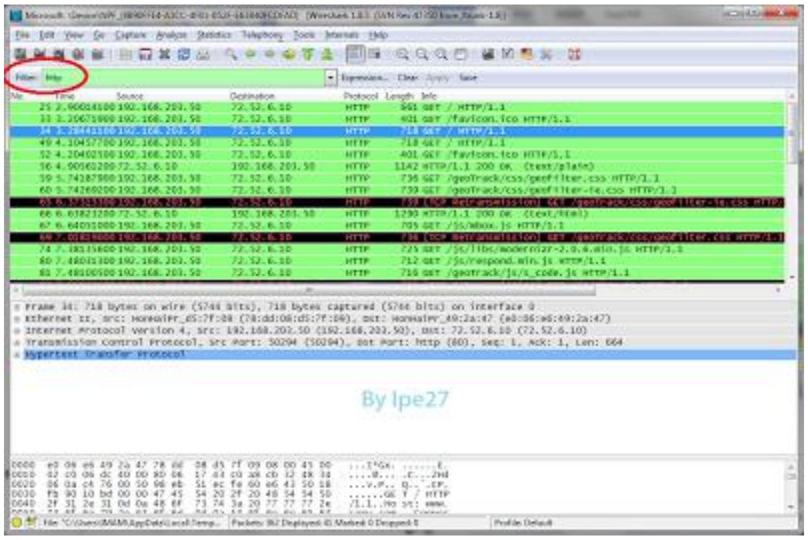
4. Setelah Web browser selesai meload Page dari situs www.sony.com, pada wireshrark lakukan stop capture (menu Capture >> Stop) maka langsung tertampil paket-paket data yang tertangkap selama meload situs www.sony.com seperti gambar dibawah.



File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
By Ipe27

```
No.  Time        Source          Destination     Protocol Length Info
 23 2.89921200  72.52.6.10      192.168.203.50  TCP       58 http > 50292 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
 24 2.89937400  192.168.203.50  72.52.6.10      TCP       54 50292 > http [ACK] Seq=1 Ack=1 Win=64400 Len=0
 25 2.90014100  192.168.203.50  72.52.6.10      HTTP     861 GET / HTTP/1.1
 26 2.90075400  72.52.6.10      192.168.203.50  TCP       58 http > 50293 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
 27 2.90086400  192.168.203.50  72.52.6.10      TCP       54 50293 > http [ACK] Seq=1 Ack=1 Win=64400 Len=0
 28 2.90125700  72.52.6.10      192.168.203.50  TCP       58 http > 50294 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1400
 29 2.90137800  192.168.203.50  72.52.6.10      TCP       54 50294 > http [ACK] Seq=1 Ack=1 Win=64400 Len=0
 30 3.13483900  72.52.6.10      192.168.203.50  TCP      809 [TCP segment of a reassembled PDU]
 31 3.13328200  192.168.203.50  72.52.6.10      TCP       54 50292 > http [ACK] Seq=808 Ack=557 Win=63843 Len=0
 32 3.16274600  192.168.203.50  72.52.6.10      TCP       54 50292 > http [FIN, ACK] Seq=808 Ack=557 Win=63843 Len=0
 33 3.20671900  192.168.203.50  72.52.6.10      HTTP     401 GET /favicon.ico HTTP/1.1
 34 3.28441100  192.168.203.50  72.52.6.10      HTTP     718 GET / HTTP/1.1
 35 3.37223200  72.52.6.10      192.168.203.50  TCP       54 http > 50292 [RST] Seq=557 Win=0 Len=0
 36 3.44041000  72.52.6.10      192.168.203.50  TCP      809 [TCP segment of a reassembled PDU]
 37 3.44085400  192.168.203.50  72.52.6.10      TCP       54 50293 > http [ACK] Seq=348 Ack=557 Win=61845 Len=0

⊞ Frame 34: 718 bytes on wire (5744 bits), 718 bytes captured (5744 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_d5:7f:08 (78:dd:08:d5:7f:08), Dst: HonHaiPr_49:2a:47 (e0:06:e6:49:2a:47)
⊞ Internet Protocol Version 4, Src: 192.168.203.50 (192.168.203.50), Dst: 72.52.6.10 (72.52.6.10)
⊞ Transmission Control Protocol, Src Port: 50294 (50294), Dst Port: http (80), Seq: 1, Ack: 1, Len: 664
⊞ Hypertext Transfer Protocol

0000  e0 06 e6 49 2a 47 78 dd  08 d5 7f 08 08 00 45 00   ...I*Gx. ......E.
0010  02 c0 06 dc 40 00 80 06  17 d3 c0 a8 cb 32 48 34   ....@... .....2H4
0020  06 0a c4 76 00 50 96 eb  51 ec fe 60 e6 43 50 18   ...v.P.. Q..`.CP.
0030  fb 90 10 bd 00 00 47 45  54 20 2f 20 48 54 54 50   ......GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 77 77 77 2e   /1.1..Ho st: www.

File: "C:\Users\[illegible]\AppData\Local\Temp\..."   Packets: 362 Displayed: 362 Marked: 0 Dropped: 0   Profile: Default
```

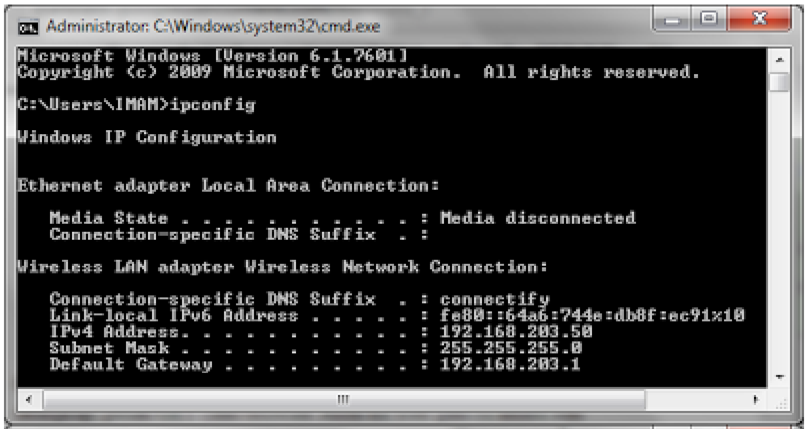
5. Karena yang akan kita analisis adalah protokol HTTP maka buatlah filter “http” pada menu filter.



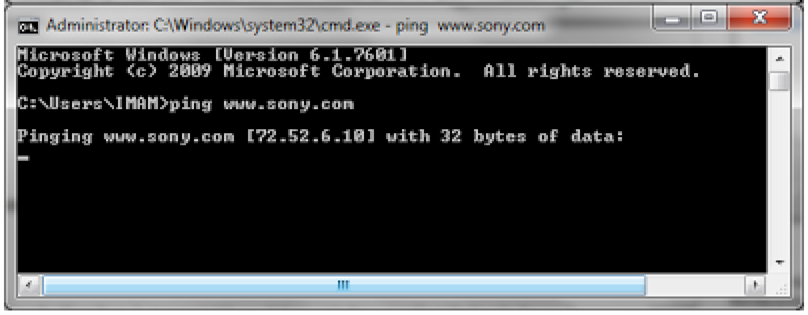
```
No.   Time        Source           Destination      Protocol Length Info
   25 2.90014100 192.168.203.50   72.52.6.10       HTTP      961 GET / HTTP/1.1
   33 3.20671900 192.168.203.50   72.52.6.10       HTTP      401 GET /favicon.ico HTTP/1.1
   34 3.28441100 192.168.203.50   72.52.6.10       HTTP      718 GET / HTTP/1.1
   49 4.10457700 192.168.203.50   72.52.6.10       HTTP      718 GET / HTTP/1.1
   52 4.20402100 192.168.203.50   72.52.6.10       HTTP      401 GET /favicon.ico HTTP/1.1
   56 4.90561200 72.52.6.10       192.168.203.50   HTTP     1142 HTTP/1.1 200 OK  (text/plain)
   59 5.74187900 192.168.203.50   72.52.6.10       HTTP      736 GET /geoTrack/css/geofilter.css HTTP/1.1
   60 5.74269200 192.168.203.50   72.52.6.10       HTTP      739 GET /geoTrack/css/geofilter-ie.css HTTP/1.1
   65 6.57323300 192.168.203.50   72.52.6.10       HTTP      739 [TCP Retransmission] GET /geoTrack/css/geofilter-ie.css HTTP/
   66 6.61823200 72.52.6.10       192.168.203.50   HTTP     1290 HTTP/1.1 200 OK  (text/html)
   67 6.64031000 192.168.203.50   72.52.6.10       HTTP      705 GET /js/mbox.js HTTP/1.1
   69 7.01829000 192.168.203.50   72.52.6.10       HTTP      736 [TCP Retransmission] GET /geoTrack/css/geofilter.css HTTP/1.1
   74 7.38135600 192.168.203.50   72.52.6.10       HTTP      725 GET /js/libs/modernizr-2.0.6.min.js HTTP/1.1
   80 7.48031300 192.168.203.50   72.52.6.10       HTTP      712 GET /js/respond.min.js HTTP/1.1
   81 7.48100500 192.168.203.50   72.52.6.10       HTTP      716 GET /geoTrack/js/s_code.js HTTP/1.1

⊞ Frame 34: 718 bytes on wire (5744 bits), 718 bytes captured (5744 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_d5:7f:08 (78:dd:08:d5:7f:08), Dst: HonHaiPr_49:2a:47 (e0:06:e6:49:2a:47)
⊞ Internet Protocol Version 4, Src: 192.168.203.50 (192.168.203.50), Dst: 72.52.6.10 (72.52.6.10)
⊞ Transmission Control Protocol, Src Port: 50294 (50294), Dst Port: http (80), Seq: 1, Ack: 1, Len: 664
⊞ Hypertext Transfer Protocol
```

Analisis :

Pada gambar diatas menunjukan bahwa banyak sekali HTTP Messages yang ditangkap saat meload situs Sony. Berdasarkan sumbernya ada dua macam HTTP Messages yang ditangkap yaitu dari browser kita ke server Sony dan sebaliknya, hal tersebut dapat dilihat dari IP source dan destination. Itu berarti kita mempunyai IP address 192.168.203.50 dan Server Sony adalah 72.52.6.10. Untuk membuktikan hal tersebut dapat dilakukan dengan menggunakan command promt seperti gambar dibawah.

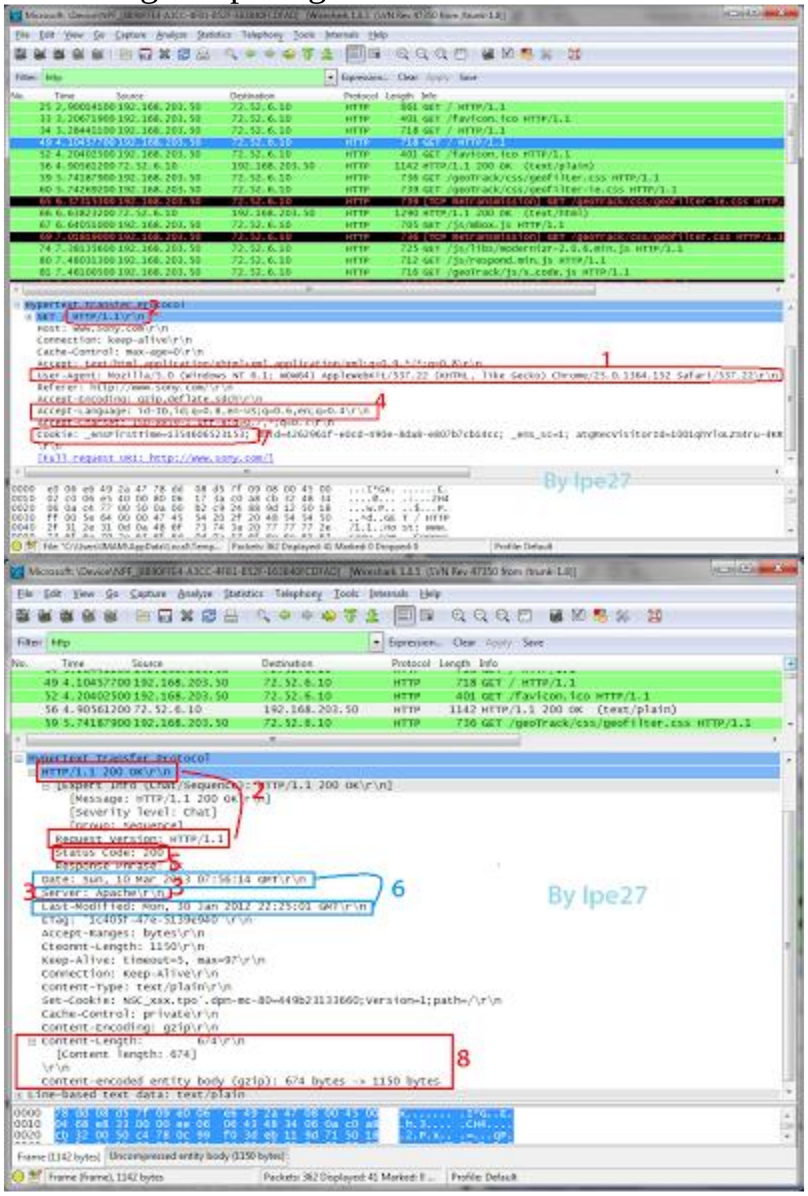


```
Administrator: C:\Windows\system32\cmd.exe

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\IMAM>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . : connectify
   Link-local IPv6 Address . . . . . : fe80::64a6:744e:db8f:ec91%10
   IPv4 Address. . . . . . . . . . . : 192.168.203.50
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.203.1
```



```
Administrator: C:\Windows\system32\cmd.exe - ping  www.sony.com

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\IMAM>ping www.sony.com

Pinging www.sony.com [72.52.6.10] with 32 bytes of data:
_
```

Untuk menganalisis protokol HTTP sebagai analogi kita buka detail dari HTTP GET dan respons messages seperti gambar dibawah.



```
No.  Time          Source            Destination       Protocol Length Info
  25 2.90014100 192.168.203.50    72.52.6.10        HTTP      861 GET / HTTP/1.1
  33 3.20671900 192.168.203.50    72.52.6.10        HTTP      401 GET /favicon.ico HTTP/1.1
  34 3.28441100 192.168.203.50    72.52.6.10        HTTP      718 GET / HTTP/1.1
  49 4.10457700 192.168.203.50    72.52.6.10        HTTP      718 GET / HTTP/1.1
  52 4.20402500 192.168.203.50    72.52.6.10        HTTP      401 GET /favicon.ico HTTP/1.1
  56 4.90561200 72.52.6.10        192.168.203.50    HTTP     1142 HTTP/1.1 200 OK  (text/plain)
  59 5.74187900 192.168.203.50    72.52.6.10        HTTP      736 GET /geoTrack/css/geoFilter.css HTTP/1.1
  60 5.74269200 192.168.203.50    72.52.6.10        HTTP      739 GET /geoTrack/css/geoFilter-ie.css HTTP/1.1
  65 6.57315100 192.168.203.50    72.52.6.10        HTTP      739 [TCP Retransmission] GET /geoTrack/css/geoFilter-ie.css HTTP/
  66 6.61823200 72.52.6.10        192.168.203.50    HTTP     1240 HTTP/1.1 200 OK  (text/html)
  67 6.64051000 192.168.203.50    72.52.6.10        HTTP      705 GET /js/mbox.js HTTP/1.1
  69 7.05819000 192.168.203.50    72.52.6.10        HTTP      736 [TCP Retransmission] GET /geoTrack/css/geoFilter.css HTTP/1.1
  74 7.38135600 192.168.203.50    72.52.6.10        HTTP      725 GET /js/libs/modernizr-2.0.6.min.js HTTP/1.1
  80 7.48031300 192.168.203.50    72.52.6.10        HTTP      712 GET /js/respond.min.js HTTP/1.1
  81 7.48100500 192.168.203.50    72.52.6.10        HTTP      716 GET /geoTrack/js/s_code.js HTTP/1.1

Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    Host: www.sony.com\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.22 (KHTML, like Gecko) Chrome/25.0.1364.152 Safari/537.22\r\n
    Referer: http://www.sony.com/\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: id-ID,id;q=0.8,en-US;q=0.6,en;q=0.4\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3\r\n
    Cookie: _enxFirstTime=1354606523153; [illegible]d=4262961f-e0cd-490e-8da8-e807b7cb4dcc; _ens_sc=1; atgRecVisitorId=100iqhYlo...
    \r\n
    [Full request URI: http://www.sony.com/]
```

```
No.  Time          Source            Destination       Protocol Length Info
  49 4.10457700 192.168.203.50    72.52.6.10        HTTP      718 GET / HTTP/1.1
  52 4.20402500 192.168.203.50    72.52.6.10        HTTP      401 GET /favicon.ico HTTP/1.1
  56 4.90561200 72.52.6.10        192.168.203.50    HTTP     1142 HTTP/1.1 200 OK  (text/plain)
  59 5.74187900 192.168.203.50    72.52.6.10        HTTP      736 GET /geoTrack/css/geoFilter.css HTTP/1.1

Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [Message: HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
  Date: Sun, 10 Mar 2013 07:56:14 GMT\r\n
  Server: Apache\r\n
  Last-Modified: Mon, 30 Jan 2012 22:25:01 GMT\r\n
  ETag: "1c405f-47e-5139e940"\r\n
  Accept-Ranges: bytes\r\n
  Cteonnt-Length: 1150\r\n
  Keep-Alive: timeout=5, max=97\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/plain\r\n
  Set-Cookie: NSC_xxx.tpo'.dpn-mc-80=449b23133660;Version=1;path=/\r\n
  Cache-Control: private\r\n
  Content-Encoding: gzip\r\n
  Content-Length: 674\r\n
    [Content length: 674]
  \r\n
  Content-encoded entity body (gzip): 674 bytes -> 1150 bytes
Line-based text data: text/plain
```

Dari data yang dicapture diatas kita dapat menganalisis beberapa informasi sebagai berikut.

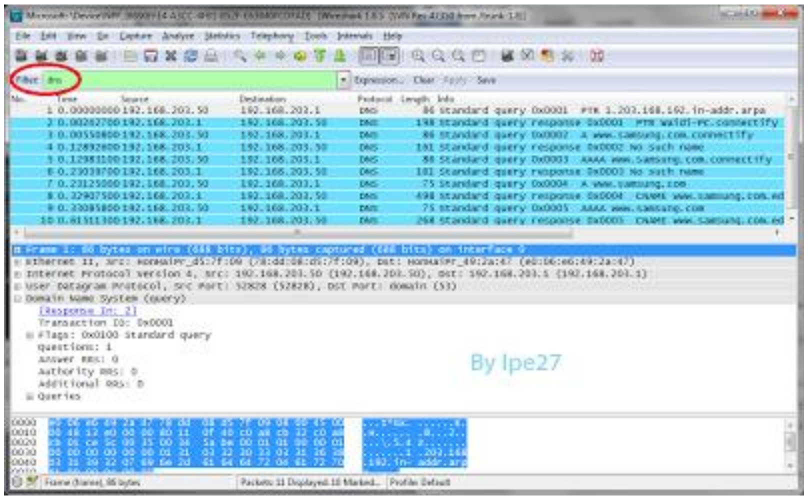
1. Kemungkinan Web browser yang digunakan adalah Mozila 5.0/Crome 25.0.1364/Safari 537.22 dengan operating sistem Windows 64bit
2. Web browser dan server Sony sama-sama running HTML versi 1.1
3. Server Sony menggunakan Apache
4. Bahasa yang digunakan Web browser yang dapat diterima server adalah bahasa Indonesia dan English US.
5. Status Code dari server ke browser kita adalah 200 yang berarti oke

6. File HTML yang kita load dari server Sony terakhir dimodifikasi server pada 30 Jan 2012 22:25:01 GMT, proses load page dilakukan pada 10 March 2013 07:56:14 GMT
7. Akses ke www.sony.com pada browser ini adalah pertama kali dan tidak ada cookie tersimpan di browser user.
8. Content yang ditransfer pada sesi ini adalah 674 bytes ----> 1150 bytes

Protokol DNS

1. Lakukan langkah 1 dan 2 seperti pada protokol HTTP diatas, namun untuk poin ketiga kita membuka situs www.samsung.com

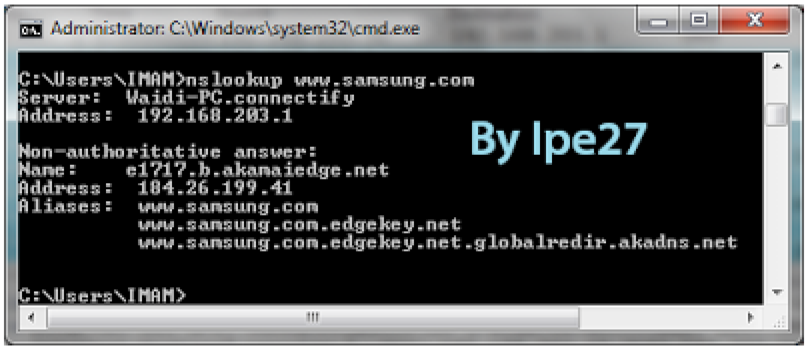
2. Lakukan capture Protokol DNS dengan membuat filter “dns” seperti gambar dibawah.



```
Filter: dns
No.  Time        Source          Destination     Protocol Length Info
  1 0.00000000 192.168.203.50  192.168.203.1   DNS       86 Standard query 0x0001  PTR 1.203.168.192.in-addr.arpa
  2 0.00262700 192.168.203.1   192.168.203.50  DNS      198 Standard query response 0x0001  PTR Waidi-PC.connectify
  3 0.00550800 192.168.203.50  192.168.203.1   DNS       86 Standard query 0x0002  A www.samsung.com.connectify
  4 0.12892800 192.168.203.1   192.168.203.50  DNS      161 Standard query response 0x0002 No such name
  5 0.12983100 192.168.203.50  192.168.203.1   DNS       88 Standard query 0x0003  AAAA www.samsung.com.connectify
  6 0.23039700 192.168.203.1   192.168.203.50  DNS      161 Standard query response 0x0003 No such name
  7 0.23125000 192.168.203.50  192.168.203.1   DNS       75 Standard query 0x0004  A www.samsung.com
  8 0.32907500 192.168.203.1   192.168.203.50  DNS      498 Standard query response 0x0004  CNAME www.samsung.com.ed
  9 0.33085800 192.168.203.50  192.168.203.1   DNS       75 Standard query 0x0005  AAAA www.samsung.com
 10 0.81511300 192.168.203.1   192.168.203.50  DNS      268 Standard query response 0x0005  CNAME www.samsung.com.ed

⊞ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_d5:7f:09 (78:dd:08:d5:7f:09), Dst: HonHaiPr_49:2a:47 (e0:06:e6:49:2a:47)
⊞ Internet Protocol Version 4, Src: 192.168.203.50 (192.168.203.50), Dst: 192.168.203.1 (192.168.203.1)
⊞ User Datagram Protocol, Src Port: 52828 (52828), Dst Port: domain (53)
⊟ Domain Name System (query)
    [Response In: 2]
    Transaction ID: 0x0001
  ⊞ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ⊞ Queries

Frame (frame), 86 bytes    Packets: 11 Displayed: 10 Marked: ...    Profile: Default
```

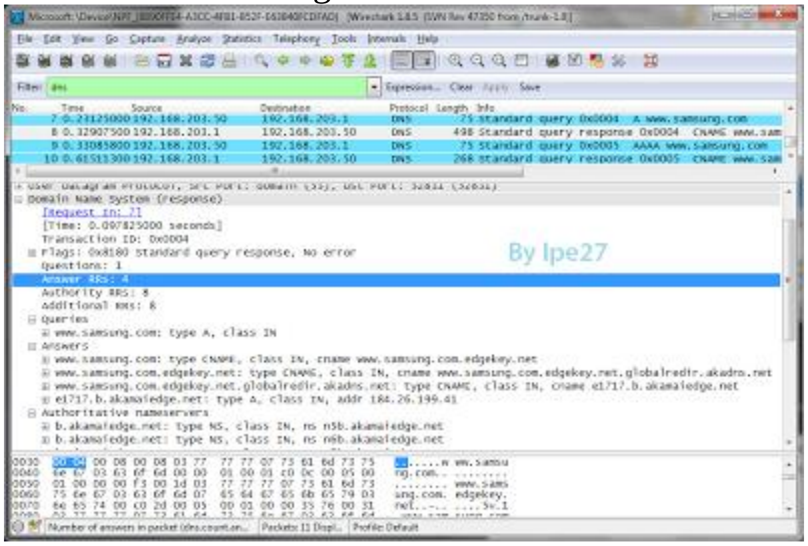
Analisis :

Dapat diketahui bahwa server DNS kita adalah Waidi-PC.connectify yang mempunyai IP address 192.168.203.1. Hal tersebut berdasar pada IP source dan destination pada penangkapan paket oleh wireshark diatas. Untuk membuktikannya kita juga dapat mengecek melalui command promt dengan command “nslookup” seperti gambar dibawah.



```
C:\Users\IMAM>nslookup www.samsung.com
Server:  Waidi-PC.connectify
Address:  192.168.203.1

Non-authoritative answer:
Name:    e1717.b.akamaiedge.net
Address:  184.26.199.41
Aliases:  www.samsung.com
          www.samsung.com.edgekey.net
          www.samsung.com.edgekey.net.globalredir.akadns.net


C:\Users\IMAM>
```

Berikut ini merupakan salah satu detail paket protokol DNS yang ditangkap wireshark saat meload situs samsung.com


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: dns
Expression... Clear Apply Save
By Ipe27
Number of answers in packet (dns.count.an...
Packets: 11 Displ...
Profile: Default

```
No.  Time        Source          Destination     Protocol Length Info
   7 0.23125000 192.168.203.50  192.168.203.1   DNS        75 Standard query 0x0004  A www.samsung.com
   8 0.32907500 192.168.203.1   192.168.203.50  DNS       498 Standard query response 0x0004  CNAME www.sam
   9 0.33085800 192.168.203.50  192.168.203.1   DNS        75 Standard query 0x0005  AAAA www.samsung.com
  10 0.61511300 192.168.203.1   192.168.203.50  DNS       268 Standard query response 0x0005  CNAME www.sam

Domain Name System (response)
    [Request In: 7]
    [Time: 0.097825000 seconds]
    Transaction ID: 0x0004
  Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 4
    Authority RRs: 8
    Additional RRs: 8
  Queries
    www.samsung.com: type A, class IN
  Answers
    www.samsung.com: type CNAME, class IN, cname www.samsung.com.edgekey.net
    www.samsung.com.edgekey.net: type CNAME, class IN, cname www.samsung.com.edgekey.net.globalredir.akadns.net
    www.samsung.com.edgekey.net.globalredir.akadns.net: type CNAME, class IN, cname e1717.b.akamaiedge.net
    e1717.b.akamaiedge.net: type A, class IN, addr 184.26.199.41
  Authoritative nameservers
    b.akamaiedge.net: type NS, class IN, ns n5b.akamaiedge.net
    b.akamaiedge.net: type NS, class IN, ns n6b.akamaiedge.net

0030  00 04 00 08 00 08 03 77  77 77 07 73 61 6d 73 75   .......w ww.samsu
0040  6e 67 03 63 6f 6d 00 00  01 00 01 c0 0c 00 05 00   ng.com.. ........
0050  01 00 00 00 f3 00 1d 03  77 77 77 07 73 61 6d 73   ........ www.sams
0060  75 6e 67 03 63 6f 6d 07  65 64 67 65 6b 65 79 03   ung.com. edgekey.
0070  6e 65 74 00 c0 2d 00 05  00 01 00 00 35 76 00 31   net..-.. ....5v.1
```

Pada dasarnya DNS (Domain Name System) adalah penerjemahan alamat IP ke hostname dan sebaliknya. Host meminta alamat yang akan dituju (www.samsung.com) kepada DNS server. Dapat dilihat untuk melakukan proses ini diperlukan 0.097825000 second.

Server DNS menjawab kepada host bahwa www.samsung.com bernama e1717.b.akmaiedge.net dengan IP address 184.26.199.41. Ternyata data ini sama persis dengan informasi yang diperoleh pada nslookup command promt.