Cybersecurity 101

A practical approach on attacking CIA triad

Devharsh Trivedi

:~# whoami

www.devharsh.me

Education

- B.E. in Computer Engineering, GTU, 2013
- M.Tech. in Information Security, Nirma, 2016
- Ph.D. student in Cybersecurity, Stevens, NJ

Work Experience

- Technical Account Manager, eClinicalWorks, Ahmedabad
- Senior Software Engineer, Philips, Bangalore
- Senior Applications Engineer, Oracle, Gandhinagar

Research Interests: Cybersecurity, Machine Learning (and Dark mode >_<)

Email me on dtrived5@stevens.edu for research collaboration (or tea party?!)

well, what else?

Blogger

- https://com.puter.tips/
- since February 2015

Member

- https://www.ieee.org/
- since October 2019

Volunteer

- https://positiveplanetus.org/
- since July 2020

Publications

- Medi Crawl - A web search engine for diseases
- R PROFILING: STRATEGIES FOR PERFORMANCE IMPROVEMENTS IN R APPLICATION
- A Study on CRAN R and MRAN R Interpreters
- Performance Evaluation of any Application with C# and R
- Near Field Communication: Overview and Applications
- Advanced WLAN Technologies: A Review

Let’s get social :)

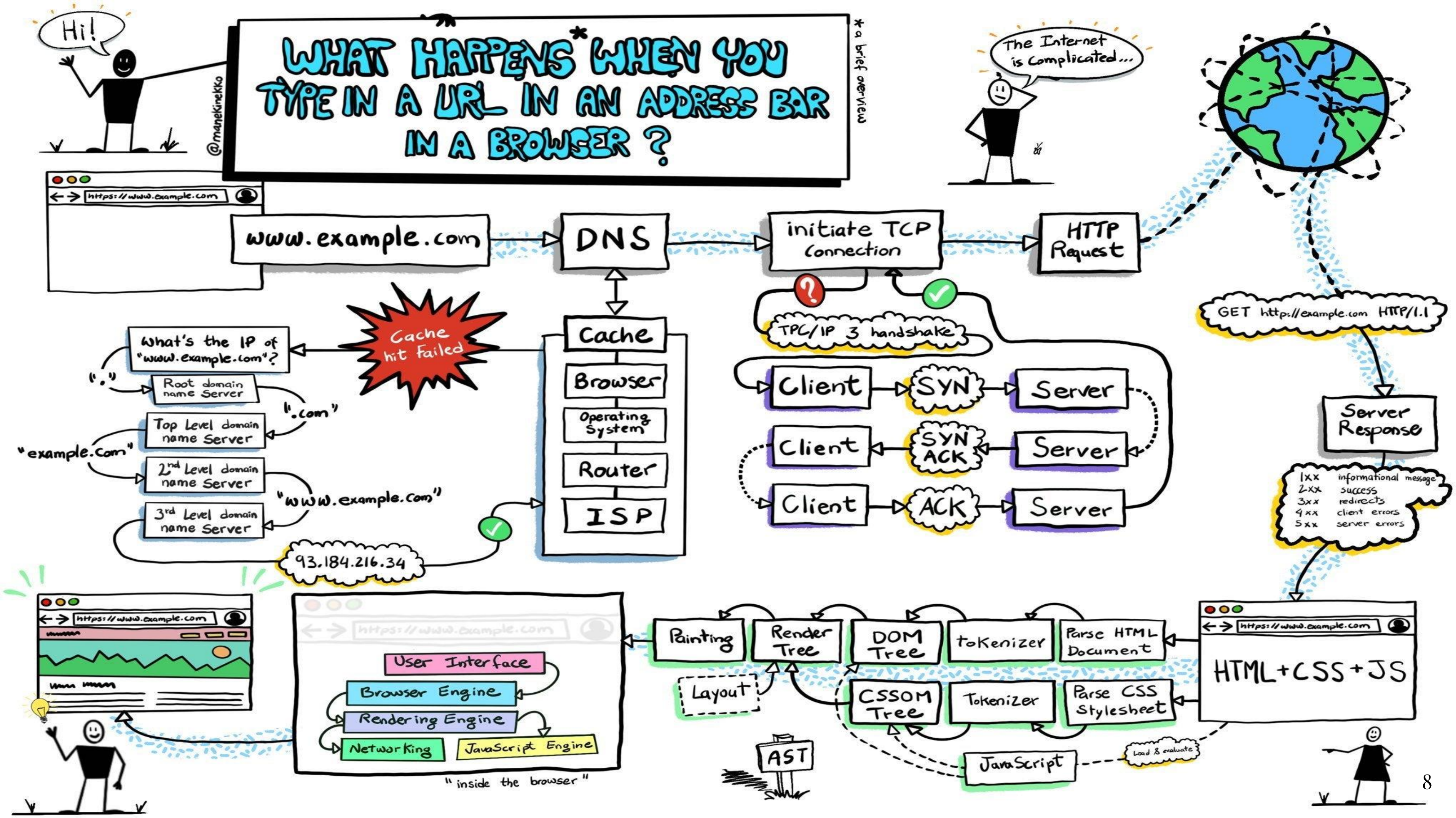
- https://www.linkedin.com/in/devharsh/
- https://www.researchgate.net/profile/Devharsh_Trivedi
- https://stackoverflow.com/users/4064166/devharsh-trivedi
- https://www.quora.com/profile/Devharsh-Trivedi
- https://github.com/devharsh

Or go to https://linktr.ee/devharsh

Cybersecurity

https://en.wikipedia.org/wiki/Computer_security

Computer security, cybersecurity or information technology security (IT security) is the protection of computer systems and networks from the theft of or damage to their **hardware**, **software**, or electronic **data**, as well as from the disruption or misdirection of the **services** they provide.



Hi!
@manekinekko
WHAT HAPPENS* WHEN YOU TYPE IN A URL IN AN ADDRESS BAR IN A BROWSER ?
* a brief overview
The Internet is Complicated...
https://www.example.com
www.example.com
DNS
initiate TCP Connection
HTTP Request
Cache hit failed
What's the IP of "www.example.com"?
"."
Root domain name Server
".com"
Top Level domain name Server
"example.com"
2nd Level domain name Server
"www.example.com"
3rd Level domain name Server
93.184.216.34
Cache
Browser
Operating System
Router
ISP
TPC/IP 3 handshake
Client
SYN
Server
Client
SYN ACK
Server
Client
ACK
Server
GET http://example.com HTTP/1.1
Server Response
1xx informational message
2xx success
3xx redirects
4xx client errors
5xx server errors
https://www.example.com
HTML+CSS+JS
Parse HTML Document
tokenizer
DOM Tree
Render Tree
Painting
Layout
Parse CSS Stylesheet
Tokenizer
CSSOM Tree
JavaScript
Load & evaluate
AST
https://www.example.com
User Interface
Browser Engine
Rendering Engine
Networking
JavaScript Engine
"inside the browser"

THE
CIA TRIAD

https://www.techopedia.com/definition/25830/cia-triad-of-information-security

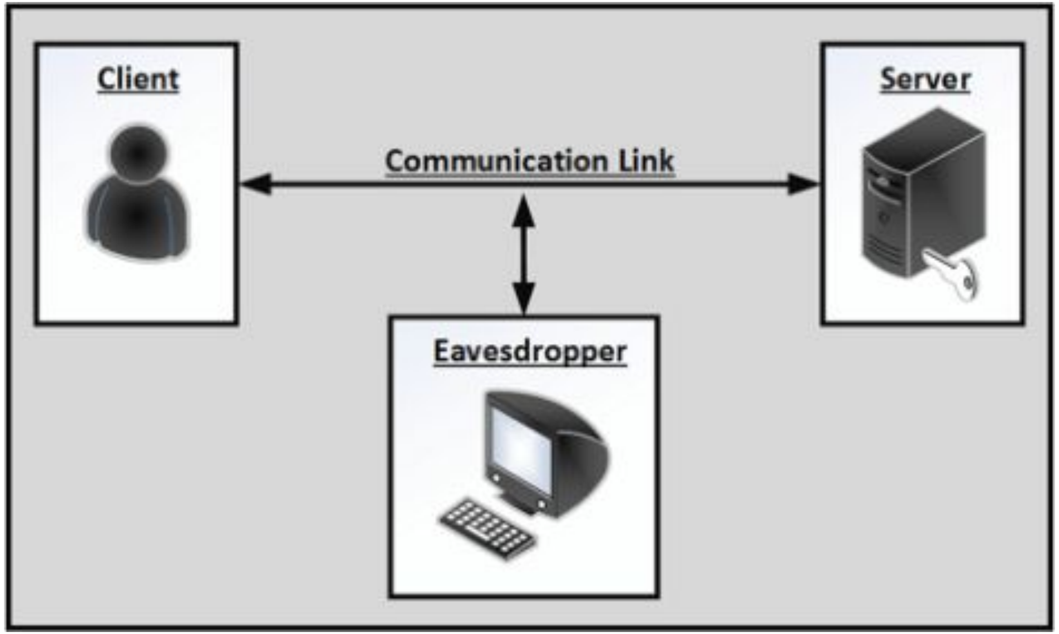
- **Confidentiality**: Ensures that data or an information system is accessed by only an authorized person. User IDs and passwords, access control lists (ACL) and policy based security are some of the methods through which confidentiality is achieved

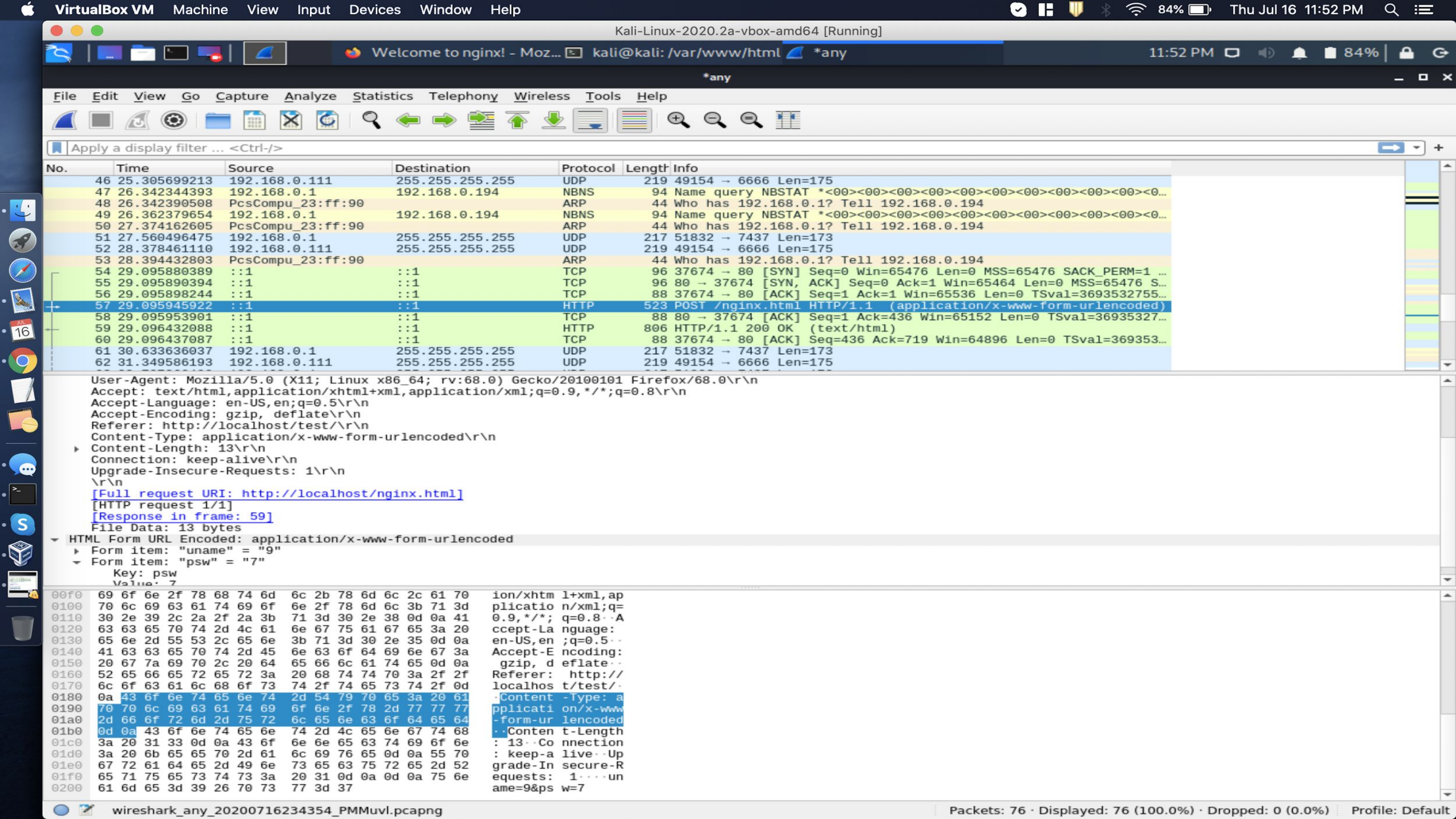
- **Integrity**: Integrity assures that the data or information system can be trusted. Ensures that it is edited by only authorized persons and remains in its original state when at rest. Data encryption and hashing algorithms are key processes in providing integrity

- **Availability**: Data and information systems are available when required. Hardware maintenance, software patching/upgrading and network optimization ensures availability

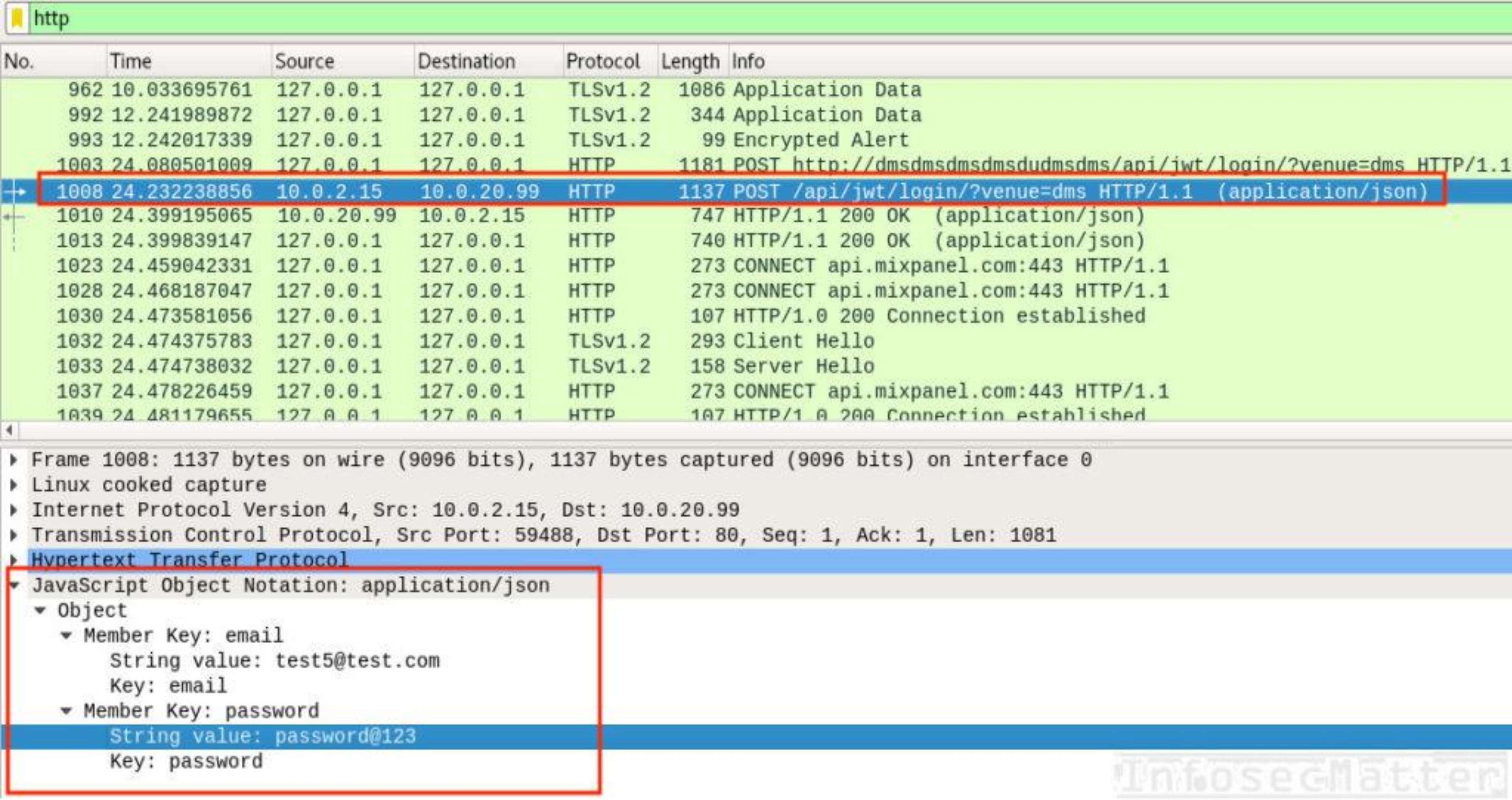
Confidentiality

Eavesdropping Attack





VirtualBox VM Machine View Input Devices Window Help
84% Thu Jul 16 11:52 PM
Kali-Linux-2020.2a-vbox-amd64 [Running]
Welcome to nginx! - Moz... kali@kali: /var/www/html *any
11:52 PM 84%
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
No. Time Source Destination Protocol Length Info
46 25.305699213 192.168.0.111 255.255.255.255 UDP 219 49154 → 6666 Len=175
47 26.342344393 192.168.0.1 192.168.0.194 NBNS 94 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><0…
48 26.342390508 PcsCompu_23:ff:90 ARP 44 Who has 192.168.0.1? Tell 192.168.0.194
49 26.362379654 192.168.0.1 192.168.0.194 NBNS 94 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><0…
50 27.374162605 PcsCompu_23:ff:90 ARP 44 Who has 192.168.0.1? Tell 192.168.0.194
51 27.560496475 192.168.0.1 255.255.255.255 UDP 217 51832 → 7437 Len=173
52 28.378461110 192.168.0.111 255.255.255.255 UDP 219 49154 → 6666 Len=175
53 28.394432803 PcsCompu_23:ff:90 ARP 44 Who has 192.168.0.1? Tell 192.168.0.194
54 29.095880389 ::1 ::1 TCP 96 37674 → 80 [SYN] Seq=0 Win=65476 Len=0 MSS=65476 SACK_PERM=1 …
55 29.095890394 ::1 ::1 TCP 96 80 → 37674 [SYN, ACK] Seq=0 Ack=1 Win=65464 Len=0 MSS=65476 S…
56 29.095898244 ::1 ::1 TCP 88 37674 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3693532755…
57 29.095945922 ::1 ::1 HTTP 523 POST /nginx.html HTTP/1.1 (application/x-www-form-urlencoded)
58 29.095953901 ::1 ::1 TCP 88 80 → 37674 [ACK] Seq=1 Ack=436 Win=65152 Len=0 TSval=36935327…
59 29.096432088 ::1 ::1 HTTP 806 HTTP/1.1 200 OK (text/html)
60 29.096437087 ::1 ::1 TCP 88 37674 → 80 [ACK] Seq=436 Ack=719 Win=64896 Len=0 TSval=369353…
61 30.633636037 192.168.0.1 255.255.255.255 UDP 217 51832 → 7437 Len=173
62 31.349586193 192.168.0.111 255.255.255.255 UDP 219 49154 → 6666 Len=175
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Referer: http://localhost/test/\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 13\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
\r\n
[Full request URI: http://localhost/nginx.html]
[HTTP request 1/1]
[Response in frame: 59]
File Data: 13 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "uname" = "9"
Form item: "psw" = "7"
Key: psw
00f0 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 ion/xhtm l+xml,ap
0100 70 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d plicatio n/xml;q=
0110 30 2e 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 0.9,*/*; q=0.8··A
0120 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 ccept-La nguage:
0130 65 6e 2d 55 53 2c 65 6e 3b 71 3d 30 2e 35 0d 0a en-US,en ;q=0.5··
0140 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a Accept-E ncoding:
0150 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 0d 0a gzip, d eflate··
0160 52 65 66 65 72 65 72 3a 20 68 74 74 70 3a 2f 2f Referer: http://
0170 6c 6f 63 61 6c 68 6f 73 74 2f 74 65 73 74 2f 0d localhos t/test/·
0180 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 61 ·Content -Type: a
0190 70 70 6c 69 63 61 74 69 6f 6e 2f 78 2d 77 77 77 pplicati on/x-www
01a0 2d 66 6f 72 6d 2d 75 72 6c 65 6e 63 6f 64 65 64 -form-ur lencoded
01b0 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 ··Conten t-Length
01c0 3a 20 31 33 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e : 13··Co nnection
01d0 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 55 70 : keep-a live··Up
01e0 67 72 61 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 grade-In secure-R
01f0 65 71 75 65 73 74 73 3a 20 31 0d 0a 0d 0a 75 6e equests: 1····un
0200 61 6d 65 3d 39 26 70 73 77 3d 37 ame=9&ps w=7
wireshark_any_20200716234354_PMMuvl.pcapng
Packets: 76 · Displayed: 76 (100.0%) · Dropped: 0 (0.0%) Profile: Default


http
No. Time Source Destination Protocol Length Info
962 10.033695761 127.0.0.1 127.0.0.1 TLSv1.2 1086 Application Data
992 12.241989872 127.0.0.1 127.0.0.1 TLSv1.2 344 Application Data
993 12.242017339 127.0.0.1 127.0.0.1 TLSv1.2 99 Encrypted Alert
1008 24.232238856 10.0.2.15 10.0.20.99 HTTP 1137 POST /api/jwt/login/?venue=dms HTTP/1.1 (application/json)
1010 24.399195065 10.0.20.99 10.0.2.15 HTTP 747 HTTP/1.1 200 OK (application/json)
1013 24.399839147 127.0.0.1 127.0.0.1 HTTP 740 HTTP/1.1 200 OK (application/json)
1023 24.459042331 127.0.0.1 127.0.0.1 HTTP 273 CONNECT api.mixpanel.com:443 HTTP/1.1
1028 24.468187047 127.0.0.1 127.0.0.1 HTTP 273 CONNECT api.mixpanel.com:443 HTTP/1.1
1030 24.473581056 127.0.0.1 127.0.0.1 HTTP 107 HTTP/1.0 200 Connection established
1032 24.474375783 127.0.0.1 127.0.0.1 TLSv1.2 293 Client Hello
1033 24.474738032 127.0.0.1 127.0.0.1 TLSv1.2 158 Server Hello
1037 24.478226459 127.0.0.1 127.0.0.1 HTTP 273 CONNECT api.mixpanel.com:443 HTTP/1.1
Frame 1008: 1137 bytes on wire (9096 bits), 1137 bytes captured (9096 bits) on interface 0
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.20.99
Transmission Control Protocol, Src Port: 59488, Dst Port: 80, Seq: 1, Ack: 1, Len: 1081
Hypertext Transfer Protocol
JavaScript Object Notation: application/json
Object
Member Key: email
String value: test5@test.com
Key: email
Member Key: password
String value: password@123
Key: password
InfosecMatter

Vigenere Cipher

- The Vigenère cipher is a method of encrypting alphabetic text by using a series of interwoven Caesar ciphers, based on the letters of a keyword.
- It employs a form of polyalphabetic substitution.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Vigenère cryptanalysis example

- **STEP 0: Capture the CipherText:**

  CTMYR DOIBS RESRR RIJYR EBYLD IYMLC CYQXS RRMLQ FSDXF OWFKT CYJRR IQZSM X

- **STEP 1: Identify n-grams:**

  CTMYR DOIBS RESRR RIJYR EBYLD IYMLC CYQXS RRMLQ FSDXF OWFKT CYJRR IQZSM X

Vigenère cryptanalysis example (cont.)

- **<u>STEP 2: Measure letters distance:</u>**
  RR-1 distance = 27 = 1 x 27, 3 x 9
  RR-2 distance = 21 = 1 x 21, 3 x 7
  CY distance = 24 = 1 x 24, 2 x 12, 3 x 8, 4 x 6
- **<u>STEP 3: Find GCD / common numbers:</u>**
  Key length = 3
- **<u>STEP 4: Divide ciphertext in 3 streams:</u>**
  CYOSSRYBDMCXRQDOKYRZX
  TRIRRIRYILYSMFXWTJIS
  MDBERJELYCQRLSFFCRQM

Crypt-analyze each stream separately

STEP 5: Count frequencies

Set 1: Red Letters -	Frequency	Set 2: Green Letters	Frequency	Set 3: Blue Letters	Frequency
C	2	T	2	M	2
Y	3	R	4	D	1
O	2	I	4	B	1
S	2	Y	2	E	2
R	3	L	1	R	3
B	1	S	2	J	1
D	2	M	1	L	2
M	1	F	1	Y	1
X	2	X	1	C	2
Q	1	W	1	Q	2
K	1	J	1	S	1
Z	1			F	2

STEP 6: Perform frequency analysis

Most frequent letters of the English alphabet: E, T, N, O, R, I, A, S

Ciphertext letter	Possible plaintext letter	Corresponded key-word letter Possible first letter of the keyword
Y	E	U
Y	T	F
Y	N	L
Y	O	K
Y	R	H
Y	I	Q
Y	A	Y
Y	S	G

Ciphertext letter	Possible plaintext letter	Corresponded key-word letter Possible second letter of the keyword
R	E	N
R	T	Y
R	N	E
R	O	D
R	R	A
R	I	J
R	A	R
R	S	Z

Ciphertext letter	Possible plaintext letter	Corresponded key-word letter Possible third letter of the keyword
R	E	N
R	T	Y
R	N	E
R	O	D
R	R	A
R	I	J
R	A	R
R	S	Z

STEP 7: Create table of possible keywords

Corresponded key-word letter Possible first letter of the keyword	Corresponded key-word letter Possible second letter of the keyword	Corresponded key-word letter Possible third letter of the keyword
U	N	N
F	Y	Y
L	E	E
K	D	D
H	A	A
Q	J	J
Y	R	R
G	Z	Z

STEP 8: Brute-force cipher

- Create all possible legal three-letters English words by choosing
  - 1st letter from the 1st column, 2nd from 2nd column and 3rd from 3rd column

- For each possible keyword, decipher the ciphertext and check its likelihood

- Possible keywords: FED, FEE, FEN, LEA, KEN, KEY, HER…

- Deciphering the ciphertext with keyword KEY will give a plaintext:

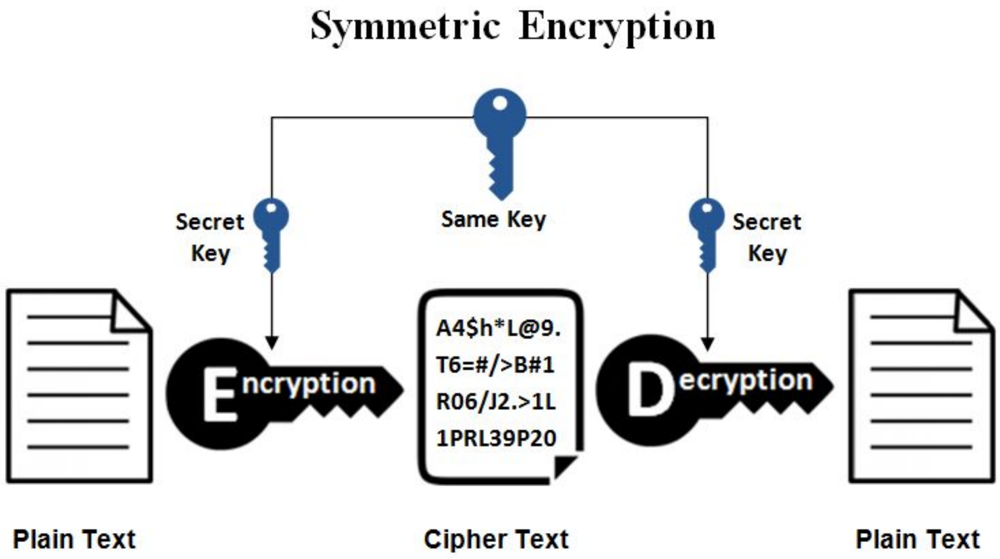
SPOON FEEDING IN THE LONG RUN TEACHES US NOTHING BUT THE SHAPE OF SPOON

Symmetric Key Encryption

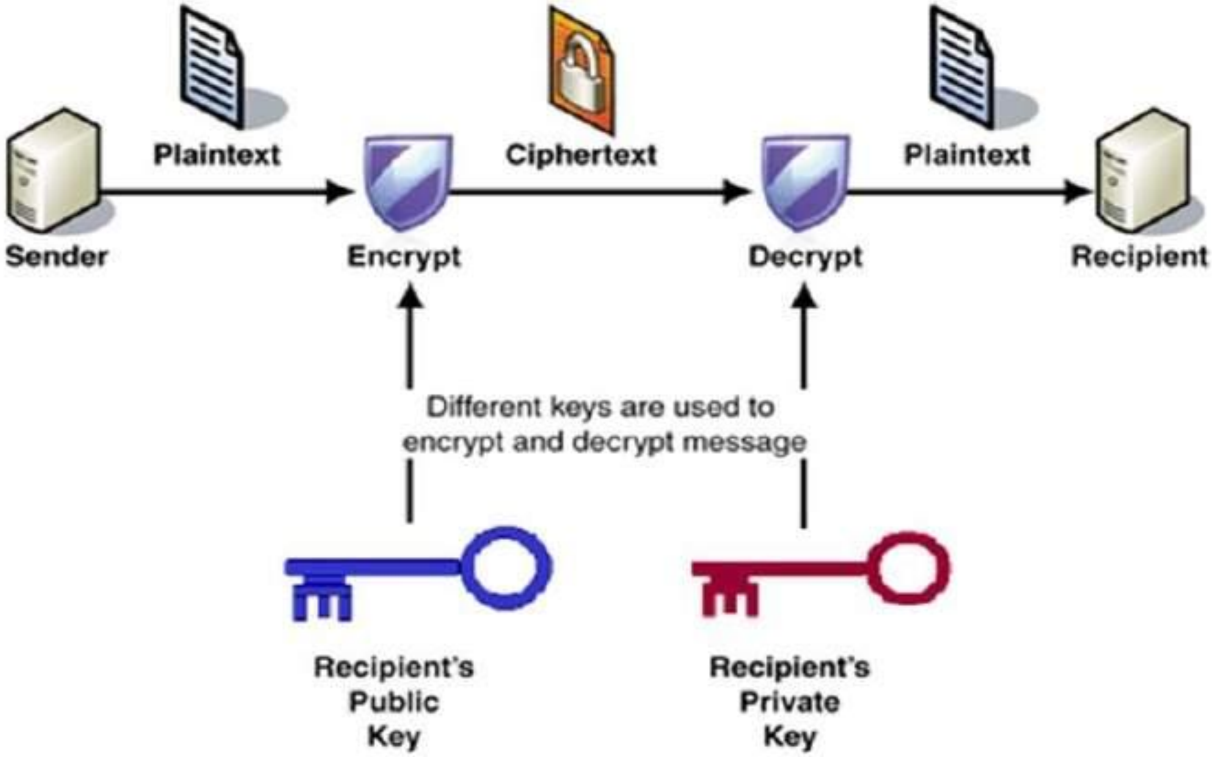
- Very fast
- Hard to crack

- Key distribution is a problem
- Do not provide other elements of security e.g., authentication, non-repudiation

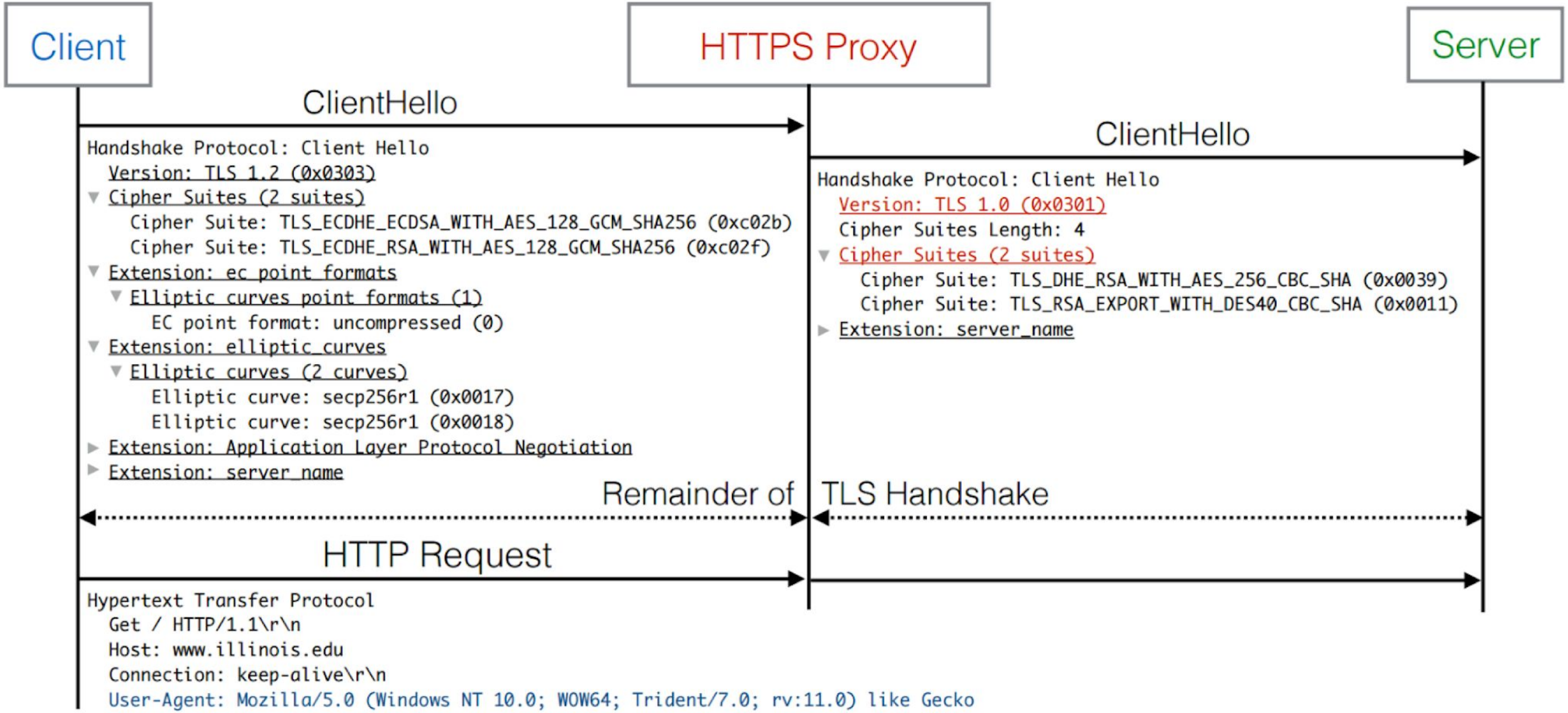
- RC4, AES, DES, 3DES, QUAD



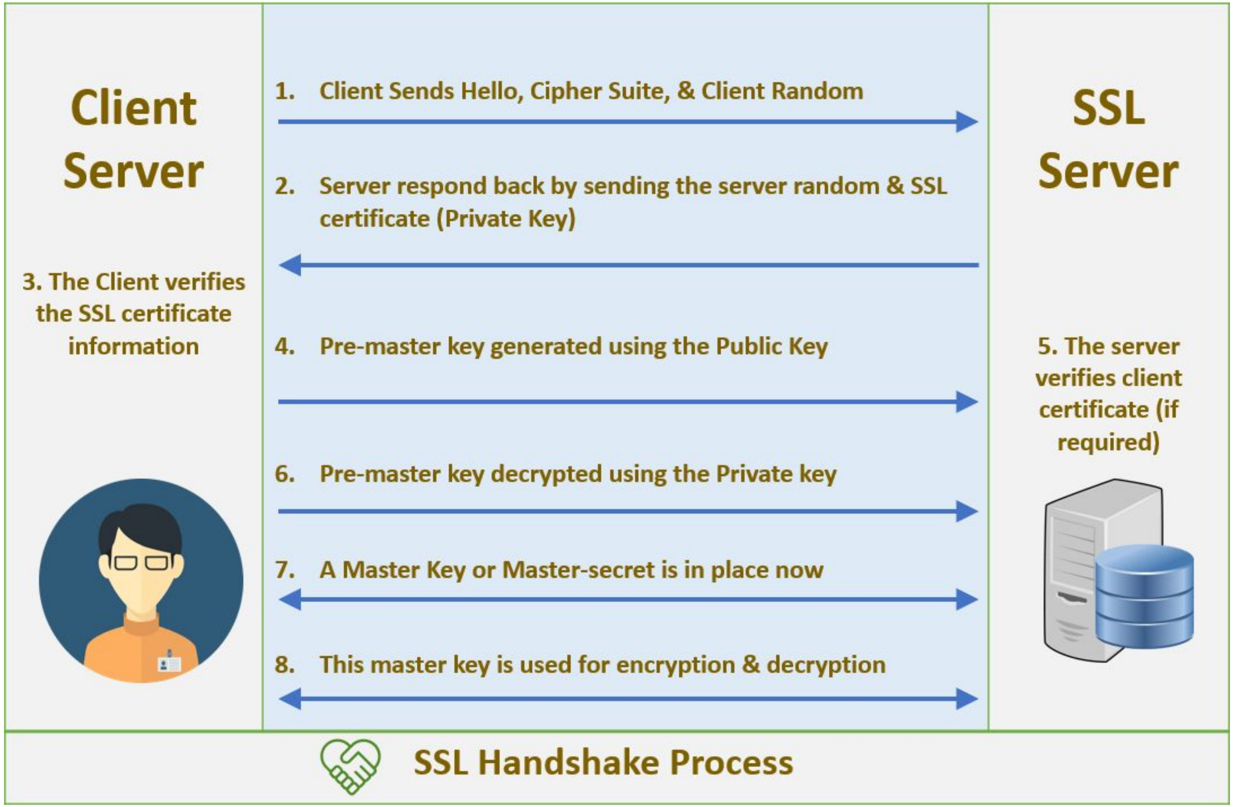
Public Key Encryption



HTTPS = HTTP + TLS(SSL)



SSL Handshake



Post Quantum Cryptography

- The problem with currently popular algorithms is that their security relies on one of three hard mathematical problems:
  - **Integer Factorization**
    - GMR, Goldwasser–Micali, Rabin, RSA
  - **Discrete Logarithm**
    - Diffie–Hellman, Elliptic-curve Diffie–Hellman, ElGamal

- All of these problems can be easily solved on a sufficiently powerful quantum computer running **Shor's algorithm**.

- Currently post-quantum cryptography research is mostly focused on six different approaches:
  - **Lattice-based cryptography**
    - **NTRU, NewHope (Ring-LWE)**
  - Multivariate cryptography
  - Hash-based cryptography
  - Code-based cryptography
  - Supersingular elliptic curve isogeny cryptography
  - Symmetric key quantum resistance

Integrity

Requirements and Security

- Most cryptographic hash functions are designed to take a string of any length as input and produce a fixed-length hash value.

- Those functions whose designs are based on a mathematical problem, their security follows from rigorous mathematical proofs, complexity theory and formal reduction.

- **Preimage:**

  For a hash value $h=H(x)$, x is the preimage of h

- **Collision:**

  A collision occurs if $x \neq y$ and $H(x)=H(y)$

Security Requirements for Cryptographic Hash Functions

- A cryptographic hash function must be able to withstand all known types of cryptanalytic attack. At a minimum, it must have the following properties:

- Preimage resistance

Given a hash h it should be difficult to find any message m such that h = hash(m). This concept is related to that of one-way function. Functions that lack this property are vulnerable to preimage attacks.

- Second pre-image resistance

Given an input m_1 it should be difficult to find another input m_2 such that $m_1 \neq m_2$ and $hash(m_1) = hash(m_2)$. Functions that lack this property are vulnerable to second-preimage attacks.

Contd.

- Collision resistance

It should be difficult to find two different messages m1 and m2 such that hash(m1) = hash(m2). Such a pair is called a cryptographic hash collision. This property is sometimes referred to as strong collision resistance. It requires a hash value at least twice as long as that required for preimage-resistance; otherwise collisions may be found by a birthday attack.

Weak hash function:

Variable input size + Fixed output size + Efficiency + Preimage resistant + Second preimage resistant

Strong hash function:

Weak hash function + Collision resistant

Collisions in MD5

- In March 2005, Xiaoyun Wang and Hongbo Yu of Shandong University in China published an [article]() in which they describe an algorithm that can find two different sequences of 128 bytes with the same MD5 hash. One famous such pair is the following:

- D131dd02c5e6eec4693d9a0698aff95c 2fcab5**8**712467eab4004583eb8fb7f89
- 55ad340609f4b30283e48883257**1**415a 085125e8f7cdc99fd91dbd**f**280373c5b
- D8823e3156348f5bae6dacd436c919c6 dd53e2**b**487da03fd02396306d248cda0
- E99f33420f577ee8ce54b67080**a**80d1e c69821bcb6a8839396f965**2**b6ff72a70

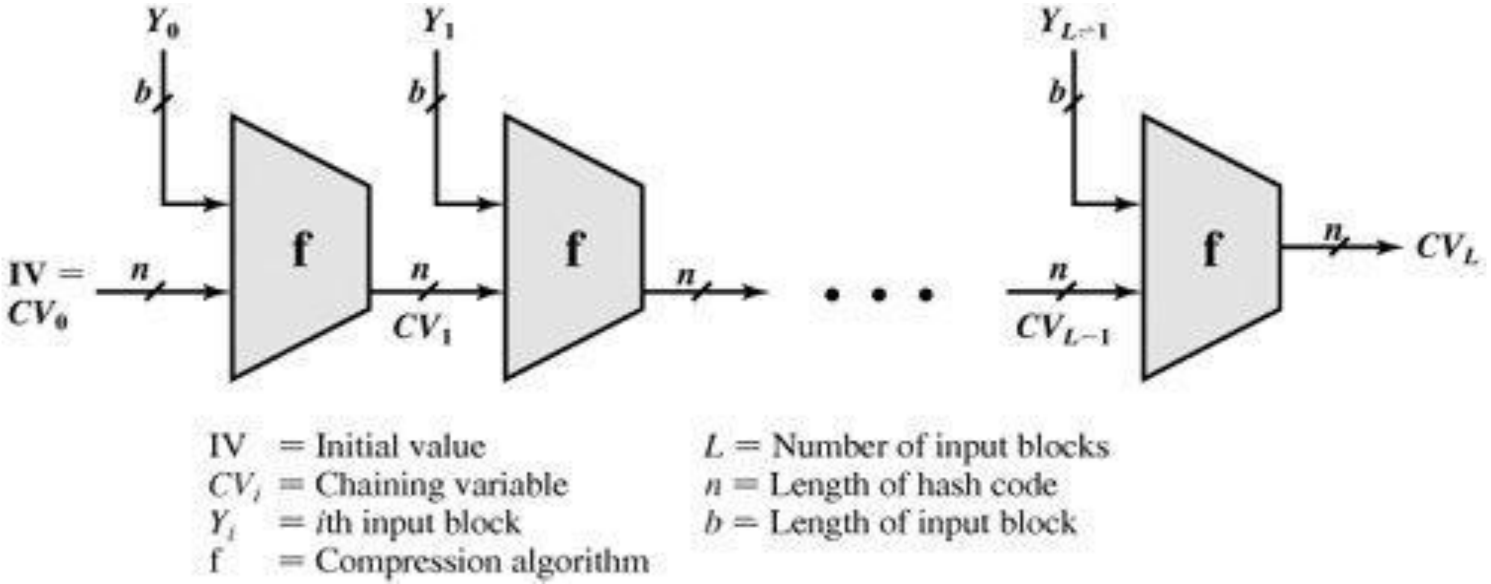
- D131dd02c5e6eec4693d9a0698aff95c 2fcab5**0**712467eab4004583eb8fb7f89
- 55ad340609f4b30283e4888325**f**1415a 085125e8f7cdc99fd91dbd**7**280373c5b
- D8823e3156348f5bae6dacd436c919c6 dd53e2**3**487da03fd02396306d248cda0
- E99f33420f577ee8ce54b67080**2**80d1e c69821bcb6a8839396f965**a**b6ff72a70

- Each of these blocks has MD5 hash **79054025255fb1a26e4bc422aef54eb4**

How Hashes are Cracked

- Dictionary attacks
- Brute Force attacks
- Lookup tables
- Reverse Lookup tables
- Rainbow tables

Cryptanalysis



Conclusion

- It is vital to have bigger key size in order to protect hash function from attacks
- Modern day hashes use more than 256 bit key length
- i.e. SHA384, SHA512
- Use salted hash
- hash("hello") = 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
- hash("hello" + "QxLUF1bgIAdeQX") = 9e209040c863f84a31e719795b2577523954739fe5ed3b58a75cff2127075ed1

Availability

Denial-of-Service (DoS) attack

- A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, **making it inaccessible** to its intended users.
- DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash.
- In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resource they expected.

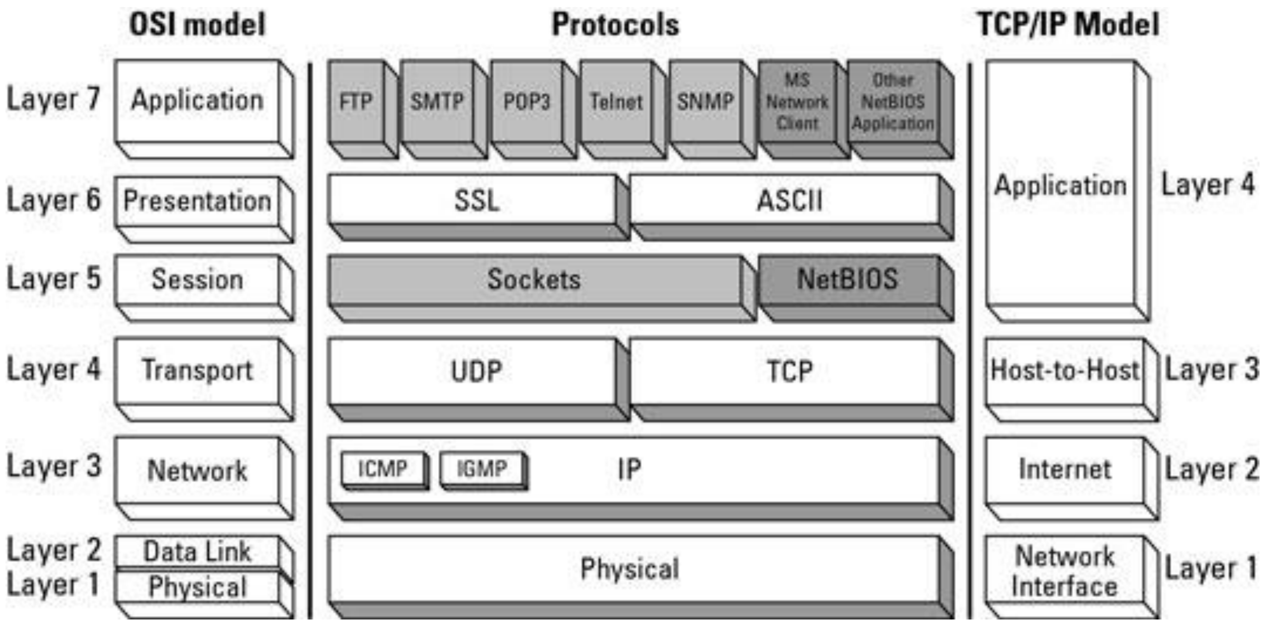
A DoS attack can happen in two ways

- **Specially crafted data (quality):** If specially crafted data is sent to the victim and if the victim is not set up to handle the data, there are chances that the victim may crash.
- **Flooding (quantity):** Sending too much data to the victim can also slow it down.

Types of the DoS attacks

Teardrop attack / IP fragmentation attack	DNS (Domain Name System) flood
UDP (User Datagram Protocol) flood	HTTP (Hypertext Transfer Protocol) flood
SYN flood	Ping (ICMP) flood
Ping of Death	Slowloris
NTP (Network Time Protocol) Amplification	SNMP (Simple Network Management Protocol) Reflection
Smurf Attack	

TCP (Transmission Control Protocol)

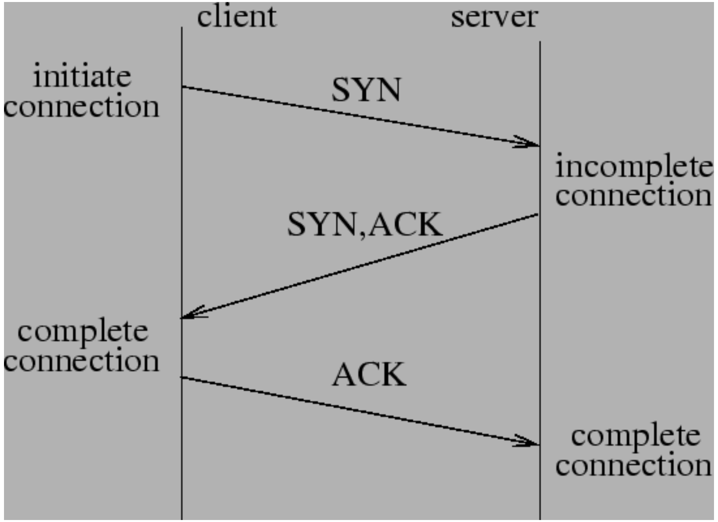


% ifconfig

Layer #	Layer Name	Protocol	Protocol Data Unit	Addressing
5	Application	HTTP, SMTP, etc…	Messages	n/a
4	Transport	TCP/UDP	Segments/ Datagrams	Port #s
3	Network or Internet	IP	Packets	IP Address
2	Data Link	Ethernet, Wi-Fi	Frames	MAC Address
1	Physical	10 Base T, 802.11	Bits	n/a

Three-way Handshake

- Client requests a connection by sending an **SYN** (synchronize) message.
- The server acknowledges by sending an **SYN-ACK** (synchronize-acknowledge) message back to the client.
- The client responds with an **ACK** (acknowledge) message, and the connection is established.



Example

Packet	***Source IP***	***Source Port***	***Destination IP***	***Destination Port***	***Seq***	***Ack***
SYN	192.168.0.166	2240	192.168.0.177	80	0	-
SYN, ACK	192.168.0.177	80	192.168.0.166	2240	0	1
ACK	192.168.0.166	2240	192.168.0.177	80	1	1

SYN flood

- The goal of an SYN flood is to tie up resources on the server machine so that it is unable to respond to legitimate connections.
- Essentially, with SYN flood DDoS, the offender sends TCP connection requests faster than the targeted machine can process them, causing network saturation.
- This is accomplished by having the **client discard the returning SYN, ACK** from the server, and not send the final ACK.
- This results in the server retaining the partial state that was allocated from the initial SYN.

LOIC (Low Orbit Ion Cannon)

hping3

- send (almost) arbitrary TCP/IP packets to network hosts
- command-line oriented TCP/IP packet assembler/analyzer
- the interface is inspired to the ping UNIX command, but hping3 isn't only able to send ICMP echo requests, it supports TCP, UDP, ICMP, and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features

% sudo hping -i u1 -S -p 80 192.168.0.177

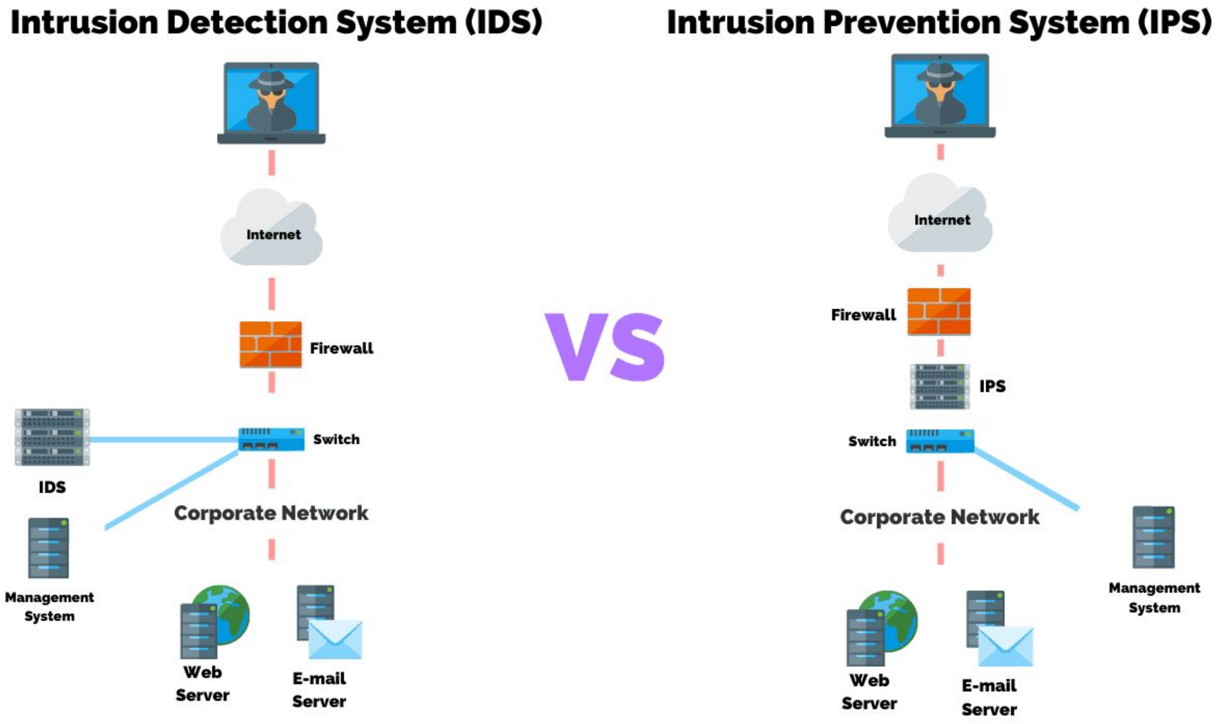
- sudo: gives needed privileges to run hping3
- hping3: calls hping3 program
- –i: –interval wait (uX for X microseconds, for example –i u1000)
- -S: specifies SYN packets
- -p 80: port 80, you can replace this number for the service you want to attack
- –flood: shoot at discretion, replies will be ignored (that's why replies won't be shown) and packets will be sent fast as possible
- -V: Verbosity
- -q: brief output
- -n: show target IP instead of the host
- -d 120: set packet size
- –rand-source: random source address mode (hide IP address)

Attack

- % ifconfig
  - active / inactive
  - ip4
- % nmap 192.168.0.160-180
- % PATH=$PATH:/usr/local/sbin
- % sudo hping -1 192.168.0.177
- % sudo hping -i u1 -S -p 80 192.168.0.177

- $ sudo wireshark
  - filter: tcp

IDS vs IPS


Intrusion Detection System (IDS)
Internet
Firewall
Switch
IDS
Corporate Network
Management System
Web Server
E-mail Server
VS
Intrusion Prevention System (IPS)
Internet
Firewall
IPS
Switch
Corporate Network
Management System
Web Server
E-mail Server

Types of IDS

- A **network** intrusion detection system (NIDS) monitors packets moving into and out of a network or subset of a network. It could monitor all traffic, or just a selection, to catch security threats.
- A **host** intrusion detection system lives on and monitors a single host (such as a computer or device). It might monitor traffic, but it also monitors the activity of clients on that computer.

Detection Methods

- **Signature-based** IDS relies on a preprogrammed list of known attack behaviors. These behaviors will trigger the alert. These “signatures” can include subject lines and attachments on emails known to carry viruses, remote logins in violation of organizational policy, and certain byte sequences.
- **Anomaly-based** IDS begins with a model of normal behavior on the network, then alert an admin anytime it detects any deviation from that model of normal behavior.

Snort

- It is an open source, free and lightweight network intrusion prevention system for Linux and Windows capable of real-time traffic analysis and packet logging.
- It can be used as a straight packet sniffer like tcpdump, a packet logger (useful for network traffic debugging, etc), or as a full-blown network intrusion prevention system.

Configuration

- $ ifconfig
- $ sudo nano /etc/snort/snort.conf
- $ snort -T -i etho -c /etc/snort/snort.conf
- $ sudo nano /etc/snort/rules/local.rules
- $ sudo snort -A console -q -c /etc/snort/snort.conf -i enpos3

Mitigation

- Increasing Backlog Queue
- Recycling the Oldest Half-Open TCP connection
- Micro blocks
- SYN cookies
- RST cookies
- Stack tweaking

Increasing Backlog Queue

- One response to high volumes of SYN packets is to increase the maximum number of possible half-open connections the operating system will allow.
- If the system does not have enough memory to be able to handle the increased backlog queue size, system performance will be negatively impacted, but that still may be better than denial-of-service.

Recycling the Oldest Half-Open TCP connection

- This strategy requires that the legitimate connections can be fully established in less time than the backlog can be filled with malicious SYN packets.
- This particular defense fails when the attack volume is increased, or if the backlog size is too small to be practical.

Micro blocks

- Allocate a micro-record (as few as 16 bytes) in the server memory for each incoming SYN request instead of a complete connection object.

SYN cookies

- The server responds to each connection request with a SYN-ACK packet but then drops the SYN request from the backlog, removing the request from memory and leaving the port open and ready to make a new connection.
- If the connection is a legitimate request, and a final ACK packet is sent from the client machine back to the server, the server will then reconstruct (with some limitations) the SYN backlog queue entry.

RST cookies

- For the first request from a given client, the server intentionally sends an invalid SYN-ACK.
- This should result in the client generating an RST packet, which tells the server something is wrong.
- If this is received, the server knows the request is legitimate, logs the client, and accepts subsequent incoming connections from it.

Stack tweaking

- Tweak TCP stacks to mitigate the effect of SYN floods.
- This can either involve reducing the timeout until a stack frees memory allocated to a connection, or selectively dropping incoming connections.

Questions??