



146 - 68KSWY7B5EIN2ST



FÖRSÄTTSLAD TENTAMEN/ EXAMINATION COVER

Jag intygar att mobiltelefon och annan otillåten elektronisk utrustning är avstängd och förvaras på anvisad plats. / I hereby confirm that mobile phones and other unauthorized electronic equipment is shut off and placed according to instructions

MARKERA MED "X" /
MARK WITH "X" ☒

IFYLLES AV STUDENT OCH TENTAMENSVAKT /
TO BE FILLED IN BY THE STUDENT AND THE INVIGILATOR:

KURSKOD / COURSE CODE D D 2 3 9 5		EFTERNAMN / FAMILY NAME KHOLIA																	
KURSNAMN / COURSE NAME Datasäkerhet		FÖRNAMN / FIRST NAME DHIRENDRA																	
PROVKOD / TEST CODE T E N 1		NAMNTECKNING / YOUR SIGNATURE 																	
TENTAMENSdatum / EXAMINATION DATE Y/Y/Y/Y M/M D/D 2 0 1 6 - 0 1 - 1 6		PERSONNUMMER / PERSONAL NUMBER Y/Y/M/M/D/D 8 5 0 2 2 7 - 8 2 5 5																	
PROGRAMKOD / PROGRAM CODE:	INLÄMNINGSTID / TIME SUBMITTED: 11:36	SIGNATUR TENTAMENSVAKT / SIGNATURE INVIGILATOR: 	ANTAL SIDOR / NO OF PAGES: 1 2																
MARKERA BEHANDLADE UPPGIFTER MED "X" OCH EJ BEHANDLADE UPPGIFTER MED "-" / MARK WITH "X" PROBLEMS SOLVED. MARK WITH "-" PROBLEMS NOT ATTEMPTED																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
X	X	X	X	X	X	X	X												

IFYLLES AV INSTITUTIONEN / TO BE FILLED IN BY THE DEPARTMENT:

BEDÖMNING / ASSESSMENT																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
V	V	V	N	V	V	V	V	V	V	V	V	V	V						

BONUSPOÄNG /
BONUS POINTS: ,

SLUTSUMMA /
FINAL POINTS: ,

BETYG /
GRADE: B

0406186500

Godkänns av examinator /
approved by Examiner.....



Family name, first name

KHOLIA, DHIRENDRA

Personal Registration Number

850227-8255

Programme

MS CS

Sheet no.

1

Problem no.

1

Sol 1.

Confidentiality loss with low impact → student grades are accidentally revealed.

Confidentiality loss with high impact → Classified national security records are revealed (Snowden style)



Family name, first name

KHOLIA, DHIRENDRA

Personal Registration Number

850227-8255

Programme

MS CS

Sheet no.

2

Problem no.

2

2a)

RSA algorithm

✓

$p, q \rightarrow$ primes (private values)

$n = p * q \rightarrow$ public value (modulus)

$\varphi(n) = (p-1) * (q-1)$

$e \rightarrow$ co-prime of $\varphi(n)$

$e * d \bmod \varphi(n) = 1 \rightarrow$ compute $(d) \rightarrow$ private value

Public key $\rightarrow [n, e] \rightarrow$ PU

Private key $\rightarrow [n, d] \rightarrow$ PR

$C = M^e \bmod n$ }
message

$D = C^d \bmod n$
 \downarrow
(m)

Problem \rightarrow Given (d, n, e, C, M) find $(e) \rightarrow$ the private exponent.

$n = p * q$ \rightarrow the mathematical equation the attacker has to solve.

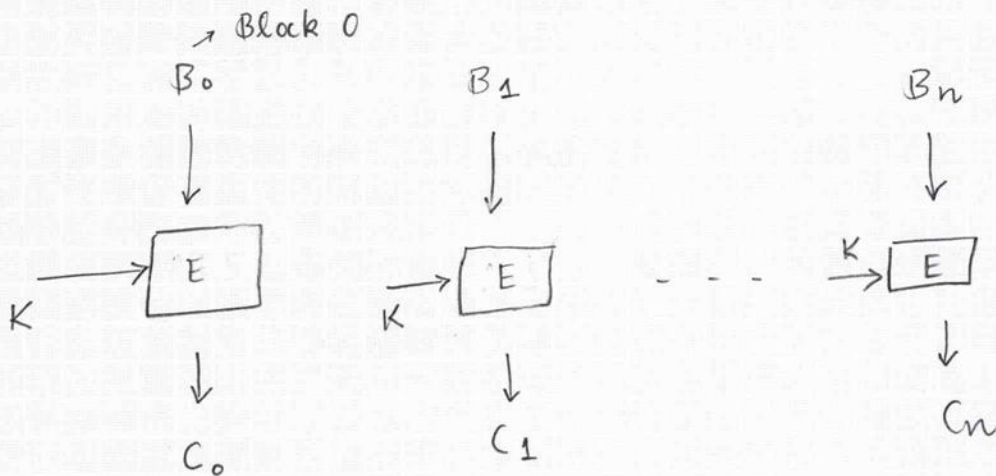
Factorization of very large integers into primes.

By finding correct (p, q) the attacker can compute (e) .

this is computationally expensive!

* Continued on the back side * !!

In ECB mode, each message block is encrypted independantly.



If the input message has repeated (similar) blocks, then the ciphertext (generated in ECB mode) will reveal the pattern of repetition.

Eg. → On the internet, there is an image of a penguin encrypted in ECB mode. The ciphertext clearly reveals the outer shape of the penguin :-)

CBC mode can be used to avoid this particular problem.



Family name, first name

KHOLIA, DHIRENDRA

Personal Registration Number

650227-8255

Programme

MS CS

Sheet no.

3

Problem no.

2

2 c)

Requirements for a secure hash function.

① one-way (preimage resistance)

$h = f(x)$, given h , it should be infeasible to find x

② collision resistant

$F(x) = F(y)$ should be infeasible to find for different messages (x, y) .

③ Handle variable length messages

Distinguish work, strong resistance.

④ Fixed size output (called message digest)

⑤ Same input should generate the same hash.

Diffusion work, strong evidence.



Family name, first name

KHOLIA, DHIRENDRA

Personal Registration Number

850227-8255

Programme

MS CS

Sheet no.

4

Problem no.

2

2d

H is not a secure hash function!

H is one-way but it is not collision resistant!

Consider,

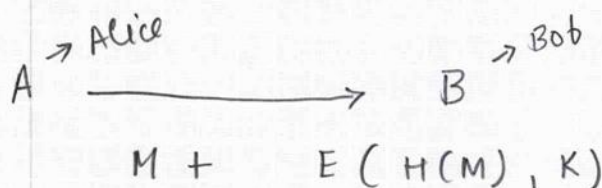
$$M_1 = B, 0, 0, 0, 0_n$$

$$M_2 = 0, B, 0, 0, 0_n$$

Both M_1 & M_2 generate same output hash!

$$H(M_1) = H(M_2)$$

High impact integrity loss \rightarrow If H is used as a MAC function (message authentication code), then we have the following problem.



\downarrow
encrypted hash \rightarrow MAC

Since H is not collision resistant, an attacker can

create M' , such that $H(M) = H(M')$ &

$$E(H(M), K) = E(H(M'), K)$$

Attacker \rightarrow B

$$M' + E(H(M'), K)$$

} \rightarrow integrity loss!

Message signature is essentially stolen, & re-used :-)

✓



Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
KHOLIA , DHIRENDRA	850227-8255	MS CS	5	3

3

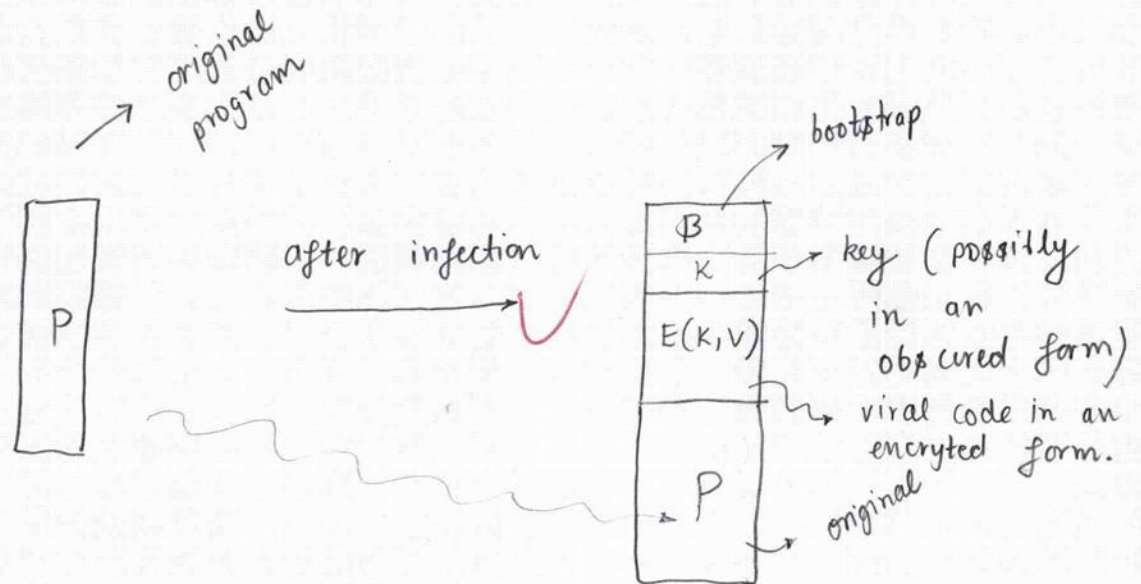
I would use a stateful inspection firewall to block the HTTP traffic that carries files bigger than 5 MB.

A simple packet filter firewall cannot inspect related HTTP requests (or their fragments) if they are split over multiple packets!

In contrast, a stateful inspection capable firewall can reconstruct the original HTTP request payload, & the corresponding HTTP responses (if required), and then block the non-allowed requests.

— x —

Solution 4



Encrypted virus infects the original program.

The execution of the infected ✓ program begins with the bootstrap code, which decrypts the encrypted viral payload by using a key embedded in the infected program itself.

The idea behind such viral constructions is to make static, even dynamic analysis of infected binaries harder. It ✓ would also be hard to develop signature-based detection rules against such virus programs.

✓



Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
KHOLTA, DHIRENDRA	850227-8255	MS CS	7	5

Solution 5 (DOS)

Example → DNS based amplified reflected Dos attack.

- ① DNS uses UDP (port 53), which implies that spoofing source IP address works well.
- ② DNS responses have the potential to be much larger than the corresponding requests. This allows the attacker to amplify his/her traffic.

The resources consumed are →



- ① Bandwidth
- ② CPU time, memory (conntrack + iptables)

Possible countermeasures →

- ① Ingress filtering (hard to do, essentially filter out attacker's traffic)
- ② Replicate, & distribute your service (servers), capacity planning
- ③ Incident response plan & ~~contingency~~ ^{don't panic!} measures
- ④ SYN cookies (in case of TCP SYN flooding based attacks).



Family name, first name

KHOLIA, DHIRENDRA

Personal Registration Number

850227-8255

Programme

MSCS

Sheet no.

8

Problem no.

6

Sol 6

Chinese Wall model is an access control model.

It is based on the "conflict of interest" idea, & essentially it prevents information disclosure between competing companies.

$O \in DS \rightarrow$ dataset
 \downarrow
object (information)

$DS \in CI$ (conflict of interest)
set

It is a pretty peculiar access control model because \rightarrow

- ① A new employee has access to only data initially.
- ② Further access is determined by which data set is first accessed by the employee.



Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
KHOLIA, DHIRENDRA Sol 6 (continued) →	850227-8255	MS CS	9	6

Chinese Wall model has following security properties →

SS-security (simple security)

- A subject can read an object (if it is the subject's initial access). Further access requests are allowed, if they aren't in conflict-of-interest control set.

*-property (star property)

- A subject can write to an object if he/she has read access to the object & he/she can't read outside the object's dataset.

The top level security property guaranteed by the Chinese wall model is confidentiality.





Family name, first name

KHOLIA, DHIRENDRA

Personal Registration Number

850227-8255

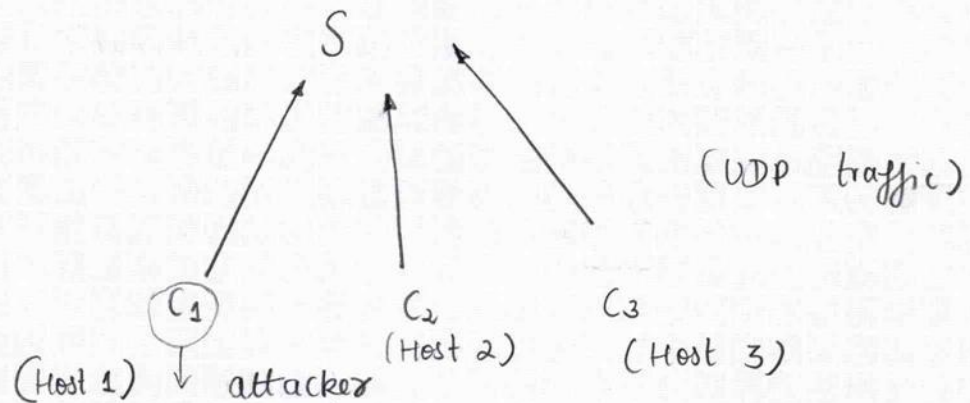
Programme

MS CS

Sheet no.

10

Problem no.

7Sol. 7 →

7a) Since the protocol is UDP based, we are able to do source IP spoofing attacks easily.

The attack (3) → The attacker (C_1, H_1) can ^(assuming they have logged on to S before!) send forged topic registration requests to the server (S) on behalf of C_2 & C_3 (Host 2, & 3) respectively.

By doing so, the attacker can put a burden on →

- a) Server resources (memory, CPU) ✓
- b) Hosts resources (memory, CPU) ✓
- c) Bandwidth & latency of the network ✓
- d) Users (are faced with spam!)

(1) victim → users (and their corresponding hosts)

(2) resource consumed → user's time! (& server + network resource consumption)

(4) assumptions needed → $(C_2, H_2), (C_3, H_3)$ --- have logged on to the server before.
+ IPs are static!

★ CONTINUED ON THE BACK SIDE ★!

(CONTINUED) ⇒

Additional security threat(s) →

- ① The server (& the people running it) know about the potentially sensitive interests of the users (based on the topics they subscribe to).

They can potentially sell this information to third-parties. It is very hard to counter this threat! ✓

②

7.6

Counter-measures

- ① Change the transport protocol from UDP to TCP.
(prevent source IP attacks from being effective)
- ② Change registration payload from $E_{PU}(\text{user}, \text{pwd})$ to

$E_{PU}(\text{user}, \text{pwd}, P_{U_{\text{user}}})$

↓
user's unique
public key

- ③ Sign the registration request payload from

"REGISTER ; topic" to "REGISTER ; topic ; signed hash of the request"

2+3 have to be done together

↓
 $E(PR_{\text{user}}, H(\text{request}))$

✓
————— × ————— × ————— × —————



Family name, first name

KHOLIA, DHIRENDRA

Personal Registration Number

850227-8255

Programme

MS CS

Sheet no.

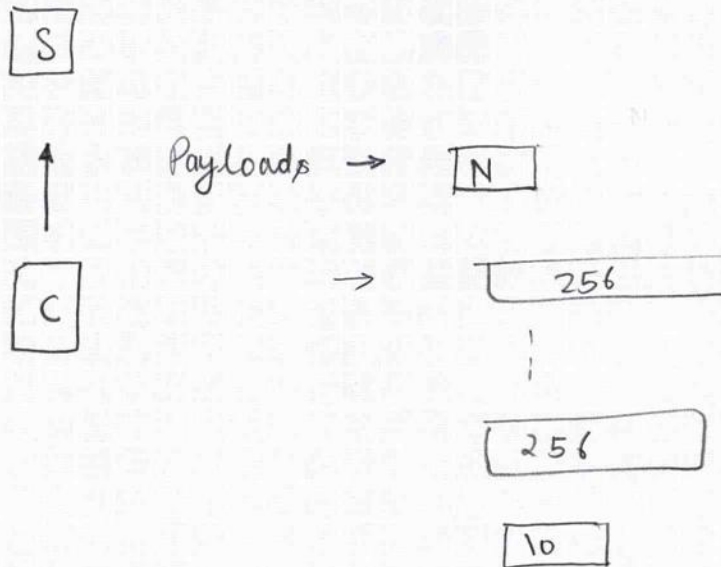
11

Problem no.

8

8

Buffer overflow



8a)

char data[n]; *user controlled!*

read(sockfd, data + (i * 256), 256); *// buffer overflow!*

{ 'n' is user-controlled (attacker-controlled), & by setting n to be smaller-than-required, the attacker can do a stack based buffer overflow attack.

Also, is "unsigned int n" always guaranteed to be at least 4 bytes wide?

If not,

read(sockfd, &n, 4); can be a bit problematic too!

architecture dependent, but ok once compiled and works ~ sort of... ✓



Family name, first name	Personal Registration Number	Programme	Sheet no.	Problem no.
<u>KHOLIA, DHIRENDRA</u> <u>86)</u>	850227-8255	MS CS	12	8

The core idea behind this fix is to make sure that we have enough space in the "data" buffer before we try to write into it.

```
while (1) {
```

```
    -  
    -  
    -  
    -
```

```
}
```



```
while (1) {
```

```
    if (n < 256)
```

```
        break;
```

```
    n = n - 256;
```

```
    read (sockfd, data + (i * 256), 256);
```

```
    if (data[i * 256] == '\0')
```

```
        break;
```

```
    i++;
```

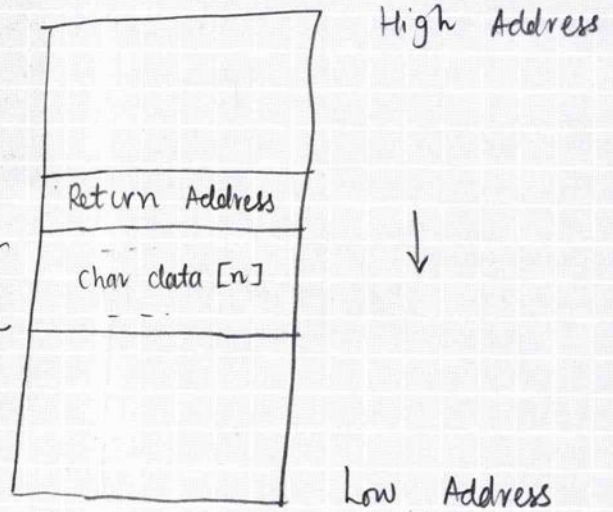
```
}
```

```
// ...
```

```
}
```


Stack Layout →
 ("function"
 is being executed)

local
 variables
 on the
 stack.

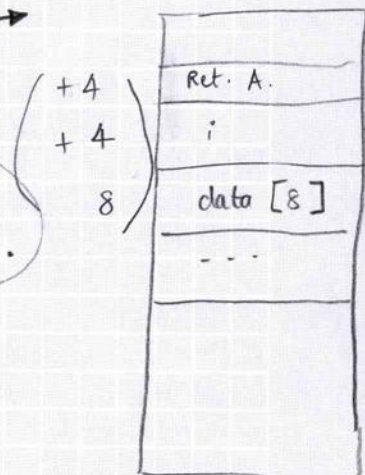


The attacker controls "n" (the size of the data buffer).
 By setting $n = 8$ (for example), the
 size of data buffer is 8.

Now the attacker can send a specially
 crafted message to overwrite the "Return
 Address" (saved on the stack) to point to
 the "nuke-world" function.

Possible message to exploit the buffer overflow →
 for ($n = 8$)

→ "A" * 12 + Ret.
 Address.
 of "nuke-world".
 (jump over "i")



The additional information that is needed
 by the attacker is the address of the "nuke-world"
 function in memory. This is easy to determine
 if the "server" binary program image is provided & the
 server doesn't use ASLR (address-space layout randomization)