

# Lab #7 - November 8, 2017

**Due** Dec 31 at 11:59pm **Points** 12 **Questions** 12

**Available** Nov 8 at 9:15am - Dec 31 at 11:59pm about 2 months

**Time Limit** None

**Allowed Attempts** Unlimited

## Instructions

**[1]** Good morning and welcome to the seventh lab session!

Lab sessions provide the opportunity for recitation and more in-depth understanding of the materials covered in class, as well as preparation for upcoming homework assignments.

Your attendance only of a given lab session (and, thus, your participation in the assignments and/or discussions) gives you full credit. You are expected to stay in the lab for the entire session, or until the TAs release the class possibly earlier than scheduled, and to actively participate in the discussions (e.g., asking questions, answering to questions, etc.).

Typically, lab assignments are offered in the form of an ungraded quiz, which should not be interpreted as a test or a mini exam.

**[2]** Today's quiz is planned so that it reviews materials covered in this week's lecture.

[Take the Quiz Again](#)

## Attempt History

	Attempt	Time	Score
LATEST	<a href="#">Attempt 1</a>	9 minutes	9.5 out of 12

Submitted Nov 8 at 1:45pm

**Question 1**

**1 / 1 pts**

## Multiplicative inverses

### Definition 1

- The residues modulo a positive integer  $n$  comprise set  $Z_n = \{0, 1, 2, \dots, n-1\}$

### Definition 2

- If  $x, y$  are two elements in  $Z_n$  and  $x$  is non-zero, such that  $xy \bmod n = 1$  then we say that " $y$  is the **multiplicative inverse** of  $x$  in  $Z_n$ " and write " $y = x^{-1}$ "

**Example:** Multiplicative inverses of the residues modulo 11 (0 does not have)

$x$	0	1	2	3	4	5	6	7	8	9	10
$x^{-1}$		1	6	4	3	9	2	8	7	5	10

## Multiplicative inverses (cont'ed)

### Fact 1

An element  $x$  in  $Z_n$  has a **multiplicative inverse** if and only if  $x, n$  are **relatively prime**

- example: the only elements of  $Z_{10}$  having a multiplicative inverse are 1, 3, 7, 9

$x$	0	1	2	3	4	5	6	7	8	9
$x^{-1}$		1		7				3		9

### Thus

If  $p$  is prime, every non-zero residue in  $Z_p$  has a multiplicative inverse

Recall the concept of multiplicative inverses that we discussed in class.

Why, whenever  $p$  is a prime, does every non-zero element  $x$  in  $Z_p$  have a multiplicative inverse?

Correct!

- ☒ Because  $p$  is a prime number.

Correct, because every such element  $x$  is, by definition, relative prime to  $p$ , since  $x$  is less than  $p$  and  $p$  is a prime.

- ☐ Because  $x$  is non-zero.

- ☐ Because  $x$  is a residue modulo  $p$ .

## Question 2

0.5 / 1 pts

### Subsets where inverses always exist

#### Definition 3

- With respect to set of residues modulo  $Z_n$ ,  $Z_n^*$  is the **subset** of  $Z_n$  containing all integers that are **relative prime** to  $n$

#### Thus

- If  $n$  is a prime, then all non-zero elements in  $Z_n$  have an inverse
  - $Z_7^* = \{1, 2, 3, 4, 5, 6\}$ ,  $n = 7$
  - $2 \cdot 4 = 1 \pmod{7}$ ,  $3 \cdot 5 = 1 \pmod{7}$ ,  $6 \cdot 6 = 1 \pmod{7}$ ,  $1 \cdot 1 = 1 \pmod{7}$
- If  $n$  is not prime, not all integers in  $Z_n$  have an inverse
  - $Z_{10}^* = \{1, 3, 7, 9\}$ ,  $n = 10$
  - $3 \cdot 7 = 1 \pmod{10}$ ,  $9 \cdot 9 = 1 \pmod{10}$ ,  $1 \cdot 1 = 1 \pmod{10}$

### Fermat's Little Theorem

#### Theorem

If  $p$  is a prime, then for each non-zero  $x$  in  $Z_p$ , we have  $x^{p-1} \pmod{p} = 1$

- Example,  $p = 5$ :

$$1^4 \pmod{5} = 1$$

$$2^4 \pmod{5} = 16 \pmod{5} = 1$$

$$3^4 \pmod{5} = 81 \pmod{5} = 1$$

$$4^4 \pmod{5} = 256 \pmod{5} = 1$$

What is the importance of Fermat's Little Theorem?

Correct!

- ☒ It provides a way to compute multiplicative inverses in  $Z_p^*$ .

Yes, a Corollary of the this theorem give us that:

If  $p$  is a prime, then the multiplicative inverse of each non-zero residue  $x$  in  $Z_p$  is  $x^{p-2} \pmod{p}$ .

Because, indeed:

$$x(x^{p-2} \pmod{p}) \pmod{p} = xx^{p-2} \pmod{p} = x^{p-1} \pmod{p} = 1$$



☐ It is not of importance for  $p$  being a prime.

orrect Answer

☐ It provides a way to compute powers more efficiently.

### Question 3

0 / 1 pts

#### RSA cryptosystem – not as used in practice!

##### Setup

RSA parameters

- ◆  $n = p \cdot q$ , with  $p$  and  $q$  primes
- ◆  $e$  relatively prime to  $\phi(n) = (p-1)(q-1)$
- ◆  $d$  inverse of  $e$  in  $\mathbb{Z}_{\phi(n)}$

Keys

- ◆ public key is  $K_{PK} = (n, e)$
- ◆ private key is  $K_{SK} = d$

##### Encryption algorithm

Encrypt

- ◆  $C = M^e \bmod n$  for plaintext  $M$  in  $\mathbb{Z}_n$

Decrypt

- ◆  $M = C^d \bmod n$

##### Signing algorithm

Sign

- ◆  $\sigma = M^d \bmod n$  for message  $M$  in  $\mathbb{Z}_n$

Verify

- ◆ return  $M == \sigma^e \bmod n$

1

The "pure" or "plain" RSA cryptosystem is described by the above exponentiation algorithms, where interestingly the same public key and secret key are used both for encryption and for signatures (of course, in reserve order). Importantly, when applied consequentially, these exponentiation algorithms have a "cancellation" property in the exponent, as explain below:

#### A useful symmetry

##### [1] RSA setting

- ◆ modulo  $n = p \cdot q$ ,  $p$  &  $q$  are **primes**, public & private keys  $(e, d)$ :  $d \cdot e = 1 \bmod (p-1)(q-1)$

##### [2] RSA operations involve **exponentiations**, thus they are **interchangeable**

- ◆  $C = M^e \bmod n$  (encryption of plaintext  $M$  in  $\mathbb{Z}_n$ )
- ◆  $M = C^d \bmod n$  (decryption of ciphertext  $C$  in  $\mathbb{Z}_n$ )

Indeed, their order of execution does not matter:  $(M^e)^d = (M^d)^e \bmod n$

##### [3] RSA operations involve exponents that "**cancel out**", thus they are **complementary**

- ◆  $x^{(p-1)(q-1)} \bmod n = 1$  (Euler's Theorem)

Indeed, they invert each other:  $(M^e)^d = (M^d)^e = M^{ed} = M^{k(p-1)(q-1)+1} \bmod n$   
 $= (M^{(p-1)(q-1)})^k \cdot M = 1^k \cdot M = M \bmod n$

2

Intuitively,  $M^{ed} = M \bmod n$  because  $M^{ed} \bmod n$  equals "1" times "M". What contributes these two factors 1 and M?

Correct Answer

☐

1 results from Euler's theorem; and M results from the fact that d, e are inverses modulo  $\phi(n)$ .

You Answered

☒

1 results from the fact that M is relative prime to n; and M results from the fact that d, e are inverses modulo  $\phi(n)$ .

☐

1 results from the fact that d, e are inverses modulo  $\phi(n)$ ; and M results from the fact that M is the base in the power  $M^{ed}$ .

## Question 4

1 / 1 pts

**RSA cryptosystem – not as used in practice!**

<p><b>Setup</b></p> <p>RSA parameters</p> <ul style="list-style-type: none"> <li>◆ <math>n = p \cdot q</math>, with <math>p</math> and <math>q</math> primes</li> <li>◆ <math>e</math> relatively prime to <math>\phi(n) = (p-1)(q-1)</math></li> <li>◆ <math>d</math> inverse of <math>e</math> in <math>\mathbb{Z}_{\phi(n)}</math></li> </ul> <p>Keys</p> <ul style="list-style-type: none"> <li>◆ public key is <math>K_{PK} = (n, e)</math></li> <li>◆ private key is <math>K_{SK} = d</math></li> </ul>	<p><b>Encryption algorithm</b></p> <p>Encrypt</p> <ul style="list-style-type: none"> <li>◆ <math>C = M^e \bmod n</math> for plaintext <math>M</math> in <math>\mathbb{Z}_n</math></li> </ul> <p>Decrypt</p> <ul style="list-style-type: none"> <li>◆ <math>M = C^d \bmod n</math></li> </ul> <p><b>Signing algorithm</b></p> <p>Sign</p> <ul style="list-style-type: none"> <li>◆ <math>\sigma = M^d \bmod n</math> for message <math>M</math> in <math>\mathbb{Z}_n</math></li> </ul> <p>Verify</p> <ul style="list-style-type: none"> <li>◆ return <math>M == \sigma^e \bmod n</math></li> </ul>
---	--

1

"Plain" RSA comprise only core algorithms that (through their useful symmetry) lend themselves to the design of an RSA public-key encryption scheme and an RSA signature scheme that are used in practice. These real-world schemes are different than plain RSA, however, in that they process (encrypt or sign) a message  $M$  not "as is," but instead as a new message  $M'$ .

That is, in practice, the RSA exponentiation functions are applied rather to new message  $M'$  that is a transformation of the original message  $M$ .

In the case of real-world RSA encryption, this message transformation includes:

### Real-world usage of RSA

- ◆ Randomized RSA
  - ◆ to encrypt message  $M$  under an RSA public key  $(e, n)$ , generate a new random session AES key  $K$ , compute the ciphertext as  $[K^e \bmod n, \text{AES}_K(M)]$
  - ◆ prevents an adversary distinguishing two encryptions of the same  $M$  since  $K$  is chosen at random every time encryption takes place
- ◆ Optimal Asymmetric Encryption Padding (OAEP)
  - ◆ roughly, to encrypt  $M$ , choose random  $r$ , encode  $M$  as  $M' = [X = M \oplus H_1(r), Y = r \oplus H_2(X)]$  where  $H_1$  and  $H_2$  are cryptographic hash functions, then encrypt it as  $(M')^e \bmod n$

31

Why  $M$  is encrypted as  $M'$  (e.g., as in Padded RSA or RSA-OAEP)?

- ☐ Because  $M$  must be relative prime to  $\phi(n)$ .
- ☐ Because  $M$  must be relative prime to  $n$ .

Correct!



Because plain RSA encryption does not provide protections against recovery of all messages, partial plaintext leakage or chosen-plaintext attacks.

Indeed, the RSA assumption states that, as long as factoring  $n$  into  $p$  and  $q$  is an infeasible task, a random message  $M$  in  $\mathbb{Z}_n^*$  cannot be inferred given  $M^e \bmod n$ . Unfortunately, this does not exclude the possibility that message recovery is possible for specific values of  $M$  or that partial information can be inferred about  $M$  (e.g., if  $M$  is even or odd number). Moreover, plain RSA encryption is deterministic, therefore it cannot possibly protect against chosen-plaintext attacks - e.g., any indistinguishability-based security notion cannot be achieved since the attacker can contrast the challenge ciphertext against the ciphertext of any of the messages  $M_1$  or  $M_2$  that he adversarially chose and provided to the challenger.



- ☐ Because smaller values of  $M$  are unsafe to be encrypted using plain RSA.

The RSA assumption states that, as long as factoring  $n$  into  $p$  and  $q$  is an infeasible task, a random message  $M$  in  $Z_n^*$  cannot be inferred given  $M^e \bmod n$ . Unfortunately, this does not exclude the possibility that message recovery is possible for specific values of  $M$  or that partial information can be inferred about  $M$  (e.g., if  $M$  is even or odd number). Moreover, plain RSA encryption is deterministic, therefore it cannot possibly protect against chosen-plaintext attacks - e.g., any indistinguishability-based security notion cannot be achieved since the attacker can contrast the challenge ciphertext against the ciphertext of any of the messages  $M_1$  or  $M_2$  that he adversarially chose and provided to the challenger.

## Question 5

1 / 1 pts

Analogously, in real-world usage of the above RSA signing algorithm, we employ the "hash & sign" paradigm:

### Digital signatures & hashing

- ◆ Very often digital signatures are used with hash functions
  - ◆ the hash of a message is signed, instead of the message itself

#### Signing message $M$

- ◆ let  $h$  be a cryptographic hash function, assume RSA setting  $(n, d, e)$
- ◆ compute signature  $\sigma = h(M)^d \bmod n$
- ◆ send  $\sigma, M$

#### Verifying signature $\sigma$

- ◆ use public key  $(e, n)$
- ◆ compute  $H = \sigma^e \bmod n$
- ◆ if  $H = h(M)$  output ACCEPT, else output REJECT

21

Why  $M$  is signed as  $M' = h(M)$  (e.g., as in RSA-FDH)?



For efficiency reasons, because, as  $M$  is typically a large integer, it helps to reduce it to a smaller value  $M'$ .

**Correct!**



Because plain RSA signatures allows for trivial forgeries.

Note that the pair  $(S^e \bmod n, S)$  is a valid message-signature pair for any value  $S$  in  $Z_n^*$ . Indeed,  $(S)^e \bmod n = S^e$  as required for a valid signature. Also, using the partial homomorphic property of the RSA function, an attacker knowing the signatures  $\sigma_1, \sigma_2$  of messages  $M_1, M_2$ , can forge signature  $\sigma_1 * \sigma_2$  for message  $M_1 * M_2$ .



For efficiency reasons, because, as  $M$  has typically large size, it helps to reduce it to a smaller value  $M'$  in  $Z_n^*$ .

Plain RSA signatures allows for trivial forgeries.

Note that the pair  $(S^e \bmod n, S)$  is a valid message-signature pair for any value  $S$  in  $Z_n^*$ . Indeed,  $(S)^e \bmod n = S^e$  as required for a valid signature. Also, using the partial homomorphic property of the RSA function, an attacker knowing the signatures  $\sigma_1, \sigma_2$  of messages  $M_1, M_2$ , can forge signature  $\sigma_1 * \sigma_2$  for message  $M_1 * M_2$ .

## Question 6

1 / 1 pts

RSA-FDH, mentioned in the previous question, is the "hash-and-sign" extension of plain RSA, where message  $M$  is signed as  $M' = h(M)$ , where  $h$  is an appropriate "full-domain" cryptographic hash function mapping messages uniformly onto  $Z_n^*$ . That is, the signature of  $M$  is  $\sigma = h(M)^d \bmod n$ , which is verified by checking whether  $\sigma^e = h(M) \bmod n$ .

How does this "hash-and-sign" extension improve plain RSA signatures?



**Correct!**

☐ Primarily, by reducing RSA to use of purely symmetric crypto-primitives.

☒ Primarily, by providing stronger security.

Indeed, hashing the message before their are being signed makes infeasible to forge signatures by solely using the public key or the partially homomorphic properties of the RSA function.

☐ Primarily, by providing better efficiency.

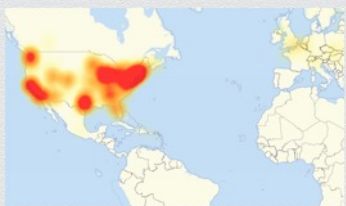
## Question 7

1 / 1 pts

It's unfair! – I had no class but couldn't watch my Netflix series!

On Friday, October 21, 2016, a large-scale cyber was launched

- ◆ it affected globally the entire Internet but particularly hit U.S. east coast
- ◆ during most of the day, no one could access a long list of major Internet platforms and services, e.g., Netflix, CNN, Airbnb, PayPal, Zillow, ...
- ◆ this was a **Distributed Denial-of-Service (DDoS)** attack



1



Please read the brief Wikipedia entry available at:

[https://en.wikipedia.org/wiki/2016\\_Dyn\\_cyberattack](https://en.wikipedia.org/wiki/2016_Dyn_cyberattack)

([https://en.wikipedia.org/wiki/2016\\_Dyn\\_cyberattack](https://en.wikipedia.org/wiki/2016_Dyn_cyberattack))

Which main security property does a Denial-of-Service (DoS) attack attempt to defeat?

☐ Integrity; services or data are modified by an unauthorized user.

☐ Confidentiality; services or data are accessed by an unauthorized user.

**Correct!**

- ☒ Availability; a user is denied access to authorized services or data.

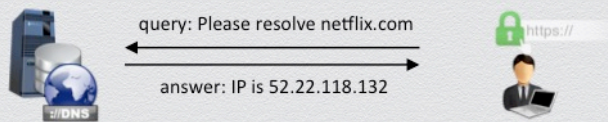
Indeed, the main goal in a DoS attack is implied by its name itself: Availability is concerned with preserving authorized access to assets and a DoS attack aims against this property.

**Question 8****1 / 1 pts**

### The Domain Name Service (DNS) protocol

Resolving domain names to IP addresses

- ◆ when you type a URL in your Web browser, its IP address must be found
  - ◆ e.g., domain name "netflix.com" has IP address "52.22.118.132"
  - ◆ larger websites have multiple IP responses for redundancy to distributing load
- ◆ at the heart of Internet addressing is a protocol called DNS
  - ◆ a database translating Internet names to addresses



2

What main security property or properties must be preserved in such an important service?

**Correct!**

- ☒ All properties in CIA triad.

Indeed, resolving domain names to IP addresses is a service that: (1) must critically be available during all times (availability); (2) must be trustworthy (integrity - or else connections to malicious sites may occur, e.g., as in a DNS-spoofing attacks); and (3) must also protect database entries that are not queried (confidentiality - or else an attacker may find out about the structure of a target organization, e.g., zone-enumeration attacks).

- ☐ Integrity.

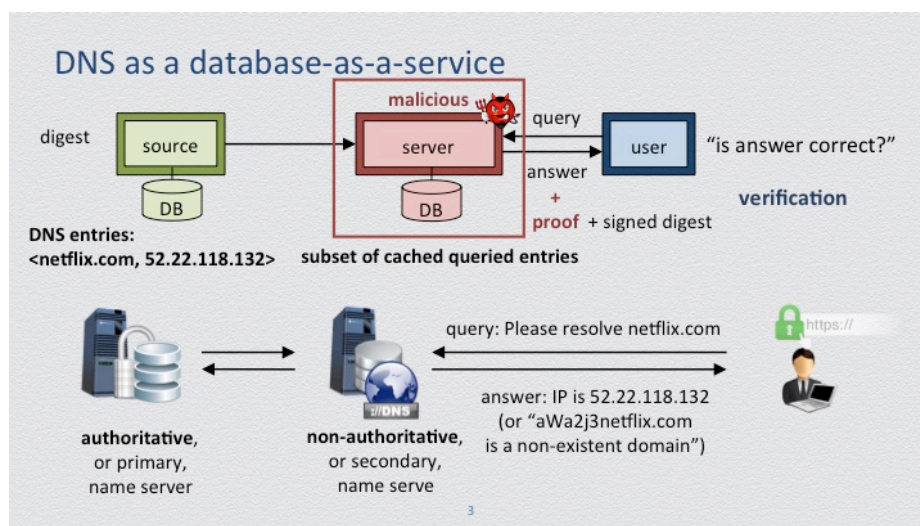
☐ Availability.

☐ Confidentiality.

All main properties of the CIA triad must be satisfied. Resolving domain names to IP addresses is a service that: (1) must critically be available during all times (availability); (2) must be trustworthy (integrity - or else connections to malicious sites may occur, e.g., as in a DNS-spoofing attacks); and (3) must also protect database entries that are not queried (confidentiality - or else an attacker may find out about the structure of a target organization, e.g., zone-enumeration attacks).

## Question 9

1 / 1 pts



As discussed in class, DNS resembles the database-as-a-service authentication model. Note that queries may have either positive verifiable answers (an existing domain name with a valid IP address, supported in DNSSEC protocol) or negative verifiable answers (an non-existing domain name with no valid IP address, also supported in NSEC) which are provided as signed "hit key-value" or "near-miss neighboring-key" pairs.



## Why DNS uses non-authoritative name servers?

Correct!

- ☒ For more scalability and locality.

Indeed, high traffic loads can saturate the response capacity of authoritative name servers. Also secondary name servers may only cache recent queried domain names without having to store large volumes of DNS entries.

- ☐ For added locality.

- ☐ For more scalability.

- ☐ For added security.

## Question 10

0 / 1 pts

**DNSSEC & NSEC**

**DNSSEC protocol:** each DNS entry is pre-signed by primary name server

**NSEC protocol:**

- domain names are lexicographically ordered and then each pair of neighboring existing domain names is pre-signed by the primary name server
- non-existing names, e.g., aWa2j3netflix.com are proved by providing this pair "containing" missed query name, e.g., <awa.com, awb.com>

4

What motivated the development of the NSEC protocol as an extension of DNSSEC and what did it result to?

Correct Answer

- ☐ Verification of negative answers & a new privacy vulnerability.

You Answered

- ☒ Lack of efficiency in DNSSEC & stronger integrity protection.

No, the goal was to support verification of non-set membership, but it resulted in introducing a new vulnerability w.r.t. the confidentiality of domain names.

- ☐ Leakage of unqueried domain names & stronger confidentiality.

## Question 11

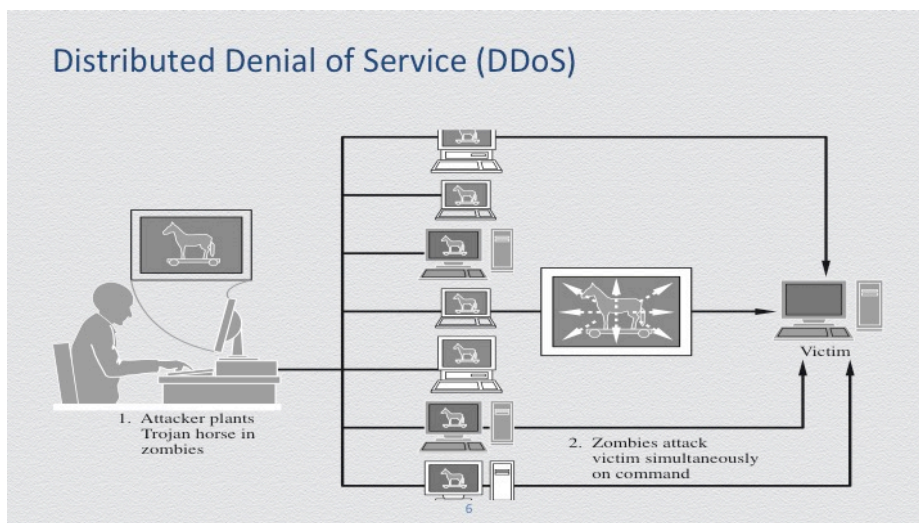
1 / 1 pts

Core idea of attack: Saturate Dyn's primary servers

**Attack:**

- from a compromised machine ask for domain names that do not exist
- query is forwarded to fewer primary Dyn servers, i.e., defeating benefits of distribution
- ask A LOT of such queries to bring down the Dyn DNS service!

5



The attack's core idea is as above. But in practice a distributed DoS attack was launched, involving employing a large army of compromised devices,

called zombies, which comprise a botnet that is controlled by the attacker, and bombarding the Dyn servers with millions of DNS queries...

Why a botnet is necessary for an effective DoS attack?

Correct!

☐ Avoid effective countermeasures.

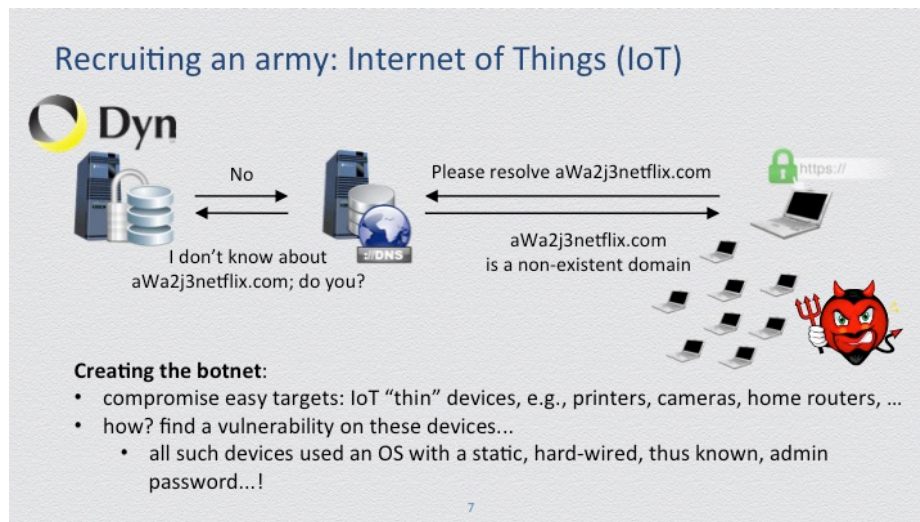
☒ Avoid effective countermeasures and increase "attack" traffic.

Yes, if the high-volume "attack" traffic comes from few devices, these devices can be filtered out by blocking their connections to the Dyn servers. Also, by employing a large botnet of million of devices the attacker inflicts a larger, more devastating "attack" traffic against the victim Dyn servers.

☐ Increase "attack" traffic.

## Question 12

1 / 1 pts



In the Dyn DDOS attack, the recruited zombie machines were IoT devices that were compromised using the [Mirai](https://en.wikipedia.org/wiki/Mirai_(malware)) ([https://en.wikipedia.org/wiki/Mirai\\_\(malware\)](https://en.wikipedia.org/wiki/Mirai_(malware))) malware.

What does the above attack method teach us?



**Correct!**

☐ Password security is an important issue.

☒ IoT security and password security are important issues.

Yes. IoT devices are "thin," not properly administrated, but they are easy targets and interconnected! Passwords are the primary user-authentication method and thus their security is crucial (as we will see next week)!

☐ IoT security is an important issue.