**ECE404 Introduction to Computer Security**

**Purdue University**

Spring 2020: Midterm-III

**Instructions**

1. Please fill-in the details on this page.
2. This is an open book, open notes exam.
3. Unless otherwise instructed, justify your answers fully.
4. **Answers that are directly copied from the lecture notes will not be accepted**.
5. **Purdue Honor Pledge: As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do.**

**Name** : Jiaxing Yang

**Student ID** : 0029284627

**Email** : yang1274@purdue.edu

**Signature** (For PDFs, use Adobe Reader’s Signature tool. For DOCX files,in Word go to Insert →Shapes→Lines →Scribble) :

# Problem 1 [11 points]

**Scenario:** Suppose you are a network security expert at a company that runs a popular ecommerce website on several servers. You are working from home on your laptop computer because of the social distancing regulations in place due to COVID-19, and your servers are located at an unknown location far away from your home. A colleague alerts you that some parts of the company’s website are taking a longer time to load than usual, and that one of the servers might be at risk of an attack. The IP address for that server is 128.46.144.100. Your goal is to figure out the cause of the slow-down as well as to resolve the problem with the server.

1. You have an account on that server with root privileges. After you enter into the server using ssh, the first thing you decide to check is if the server is under SYN-Flood attack. To do that you plan to use netstat utility. After you run the appropriate command, you get the following output:

tcp 0 0 128.46.144.100:80 140.53.84.24:609 SYN RECV

tcp 0 0 128.46.144.100:80 121.122.42.52:13 SYN RECV

tcp 0 0 128.46.144.100:80 129.12.192.16:48 SYN RECV

tcp 0 0 128.46.144.100:80 121.132.55.21:17 SYN RECV

tcp 0 0 128.46.144.100:80 121.152.212.2:17 SYN RECV

tcp 0 0 128.46.144.100:80 121.21.412.12:13 SYN RECV

tcp 0 0 128.46.144.100:80 124.18.11.10:317 SYN RECV

Observing the output above, you conclude that the server is definitely under a SYN-Flood attack. What would prompt you to reach that conclusion? **[3 pts]**

The main reason is that the lines are repeating in the SYN RECV and never get out of the state, which means the machine is stuck at the SYN RECV state when the packets received.

1. You attempt to thwart the attack by blocking the IP addresses mentioned in the 5th column (i.e., the column that shows the IP addresses of the remote hosts) of the output shown in part (a). But this is not a good strategy. Why? **[3 pts]**

The reason why this is not a good strategy is because the attacker can easily spoof their IP addresses, if you block this IP address, they can just change the IP address to others and still do the attack.

1. You then decide to limit the request for a new connection to one per second. What command would you use to accomplish this? [Hint: use a packet filtering firewall] **[5 pts]**

The command will be:

iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT

# Problem 2 [12 points]

1. The DNS hierarchy includes root servers, gTLD servers, and ccTLD servers. What role do each of these servers play when a client first contacts the system? **[5 points]**

When a client first contacts the system, the root servers will look up and return the IP address of gTLD servers or ccTLD servers. If the root servers receive a query for the generic domain, for example, those ones with “.com”, “.edu”, they root server will sends back the IP address of one or more gTLD nameservers in charge of those domain. If the root servers receive country-code top-level domain, for example, those with “.uk”,”.jp”, the root server will return the IP address of one or more ccTLD nameservers.

1. What is DNS cache poisoning? How is it done? What aspect of DNS makes it a difficult attack to pull off? **[7 points]**

The DNS cache is poisoning the cache by entering a fake IP address for a hostname, a domain name or another nameserver.

This will be done by sending corrupt data to the cache of the real DNS name server. Then the corrupt data(containing the fake IP address) will be stored as the query result of the local nameserver, so the local nameserver will return the fake IP address to the user, and the user will enter the fake domain as the result.

The aspect of DNS makes it a difficult attack to pull of is the use of a 16-bit transaction ID integer that is sent with every DNS query, because the transaction ID is randomly generated.

# Problem 3 [9 points]

1. Let’s say I have five computers in a LAN functioning as a server and I want to designate one computer for protecting the LAN with a firewall that will direct traffic destined for port 22 to the other 4 machines . What is the general term for this process? What would be the iptables command for accomplishing this if the IP address of the firewall computer is 128.210.107.65 and the other 4 machines have local IP addresses 10.0.0.1-10.0.0.4? **[5 points]**

The general term for this process is called port forwarding.

The iptables command for accomplishing this if the IP address of the firewall computer is 128.210.107.65 and the other 4 machines have local IP addresses 10.0.0.1-10.0.0.4 are:

iptables -dport 22 -t nat -A PREROUTING -p tcp -d 128.210.107.65 -j DNAT --to-destination 10.0.0.1-10.0.0.4

1. Why does the usage of salt in a password hashing scheme make it difficult for an attacker to mount a dictionary attack, even if the attacker knew the salts that were used? **[4 points]**

The reasons why the usage of salt in a password hashing scheme make it difficult for an attacker to mount a dictionary attack are below:

1. The salt is randomly chosen bit pattern, and it is combined with the actual password, then the combination will be hashed by a hashing algorithm.
2. Even when an attacker stole the files in /etc/shadow, and they know the salt used for each username, they will not be able to use pre-computed rainbow tables in order to crack the passwords.
3. It will take years for attackers to create their rainbow tables to match every possible value of salt (hard to break).

# Problem 4 [16 points]

1. Let’s say you are purchasing something on eBay.com. What role does TLS/SSL play in ensuring this transaction is secure? **[4 points]**

TLS/SSL plays a central role in the process, and here are the explanations:

1. During the purchasing process, before entering the credit card information, TLS/SSL protocol will make sure the remote host is real so that the information of the credit card can be sent.
2. Explain at a high level what is accomplished at each phase of an SSL Handshake between two parties for the SSL Handshake protocol. By high level, we mean enough to understand what the two parties are doing in each phase without the need to be precise about the parameters they exchange. **[7 points]**

There are four phases in the SSL Handshake protocol, here are the details of each phase.

1. Phase 1 is used to establish the security capabilities for both server and client. In this phase, the client sends to the server a client\_hello message including version, random, session ID, Cipher Suite and Compression Method.
2. Phase 2 is initiated by the server by sending the server certificate to the client. Here are the rest of the actions:
   1. Server sends to the client the message labeled certificate. This certificate containing its one or more certificates, where they are used to valid the server’s public key.
   2. A message containing the server\_key\_exchange message and a certificate\_request message will follow the certificate message.
3. Phase 3 steps:
   1. The client first sends the client’s certificate to the server.
   2. Then, the client sends to the server a message called client\_key\_exchange message.
   3. At last, the client sends to the server a message called certificate\_verify message to verify whether the certificates are signed by a certificate authority.
4. Phase 4 completes setting up the connection between the client and the server via the following steps:
5. Client sends to the server a change\_cipher\_spec message, which indicates that it is copying the pending CiperSpec into the current CipherSpec
6. The client sends to the server the finished message.
7. Server does the same vis-à-vis the client.

3. Explain the relationship between a socksified client, a SOCKS server, and the outside internet (i.e. how does each of these three entities interact with each other?) **[5 points]**

The relationship between a socksified client, a SOCKS server, and the outside internet is as the following:

1. The socks client wraps all the network-related system calls made by a host with its own socket calls, this is referred as socksifying the client call.
2. The socks server forwards the request to the server on the internet after the socks server request made by the socksified client for its legitimacy.
3. Any response received back from the server is forwarded back to the LAN client.

# Problem 5 [5 points]

1. Assume that a host A decides to use the TOR protocol to send a message anonymously to host E on the internet. The TOR protocol constructs the following circuit from host A to host E:

A → B → C → D → E

Here, nodes B, C, and D are the onion routers (ORs) of the TOR overlay.

While sending messages to nodes in the circuit, each node does not need to share its public RSA key with subsequent nodes in the circuit (e.g. node A does not share its public RSA key with any other nodes, B does not share its key with nodes C, D, and E, node C does not share its key with nodes D and E, etc.).

Explain what makes it possible for TOR to securely transmit messages without a node needing to share its public RSA key with subsequent nodes in the circuit. **[5 points]**

Here are the reasons make it possible for TOR to securely transmit messages without a node needing to share its public RSA key with subsequent node in the circuit:

1. The data field will contain the DH public key to the specific OR when A sends replay packet to each of the ORs.
2. DH public key during the A ->E process will be encrypted by K\_AB, K\_AC, K\_AD, K\_AE, and from A to B, the message only decrypted with K\_AB, however, the message is still encrypted with K\_AC, K\_AD and K\_AE, therefore, B is not able to see the message and there is no need to share its public RSA key. Similarly speaking, decrypt each time till reaching E, the message will be lastly decrypted with K\_AE and the previous ORs are not able to see the message and there is no need to share the RSA public key.

# Problem 6 [20 points]

1. Is the following statement True or False? “In a C program, if I use a malloc call in a declaration of a buffer, that buffer cannot overflow.” Please justify your answer. **[4 points]**

The statement is False, because buffer overflow can happen in a heap, where malloc allocates in heap as well. Buffer overflow in heap happens when the size of information written out to a memory location exceeds the block of memory allocated for the object at that location, the overwrite in the adjoining memory locations can corrupt the data.

1. Let’s say a canary with a value of ***0xc69b05c8*** is used to detect if a buffer overflow attack has occurred in the execution of a program. Would an attacker still be able to mount a buffer overflow attack despite the presence of this canary? Explain why or why not. **[4 points]**

No, because any change in the guard value stored in canary taken as an attempt at buffer overflow exploitation. During the buffer overflow attack, the attacker will have to change the overflow string to incorporate with the canary value, however, the strcpy() and gets() function will not work. If the attacker tries to change the value at canary, the epilogue would detect and abort the process.

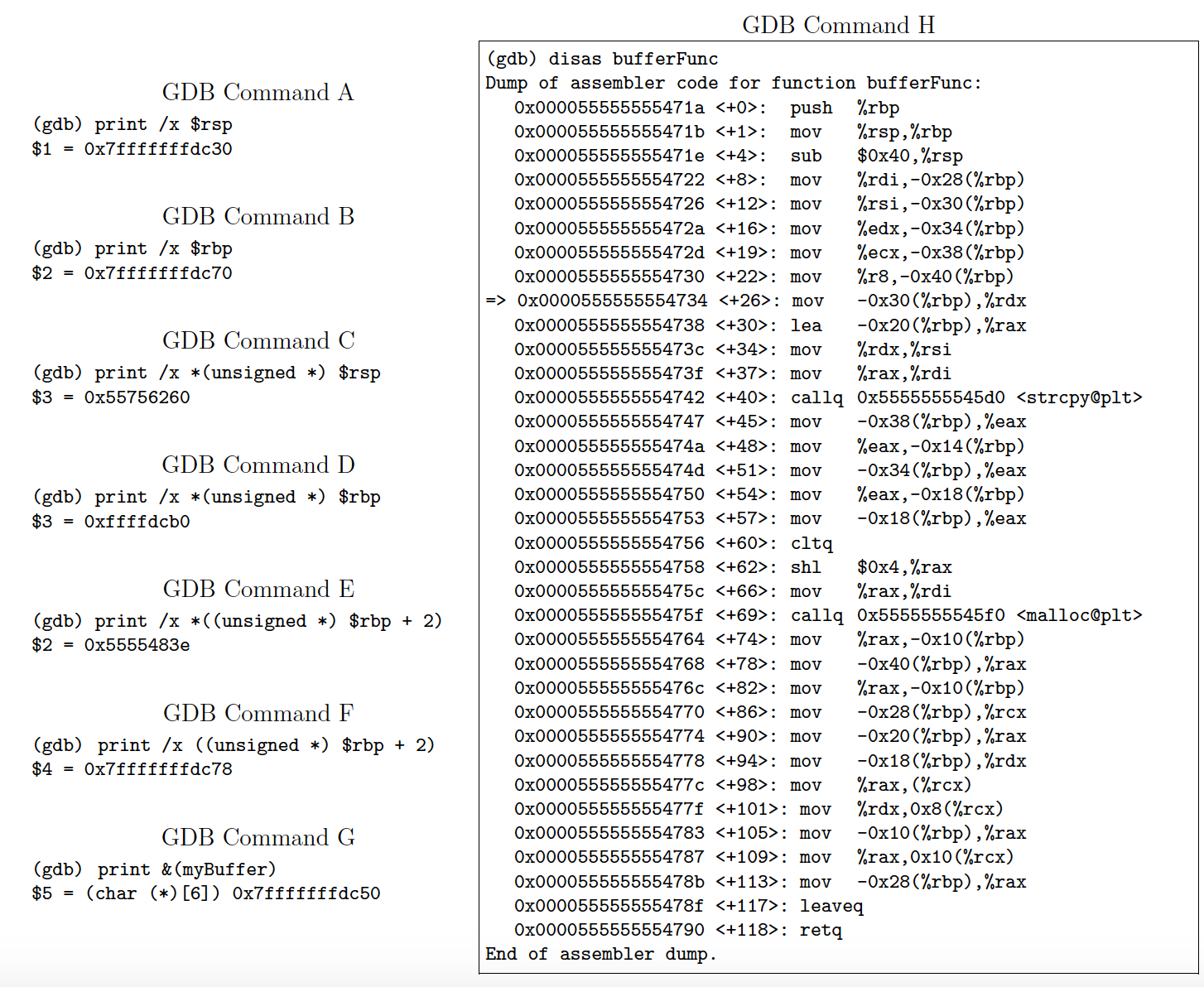
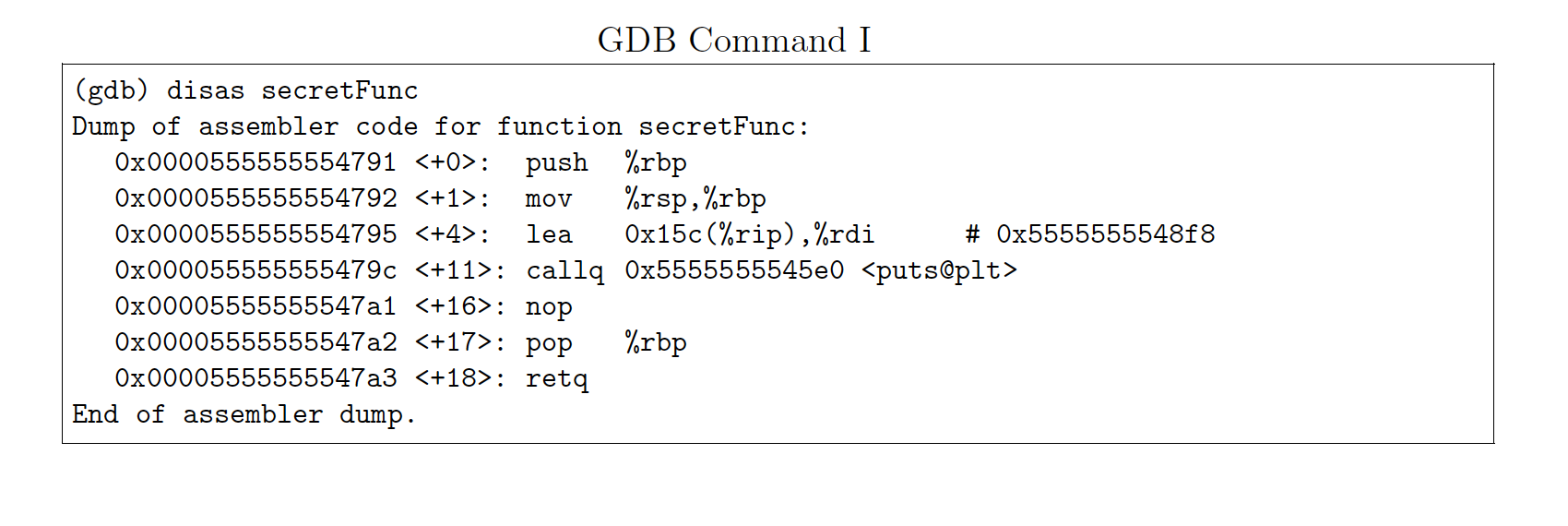
1. Let’s say there is a C program with a buffer overflow vulnerability in the function *bufferFunc(...)* with the variable *char myBuffer[6]* due to the use of the *strcpy(...)* function to copy user input into *myBuffer*. An attacker wants to exploit this vulnerability to execute the function *secretFunc(...)*. In order to craft a string to mount the attack, the program is run with **gdb**. On the following pages are some **gdb** commands along with their respective outputs at a breakpoint right before the vulnerability (you are allowed to use a Hexadecimal to Decimal converter for the addresses). Using this information, create a string that could potentially be used to enter *secretFunc(...).* Be sure to explain how you came up with the string and refer to the GDB commands that helped you craft the string. **[12 points]**

Here are the steps to find out the string:

1. According to the graph the first memory location that signifies the entry into the object code for secretFunc() is 55554791 (we only consider the last 4 bytes), therefore, we can deternine the last four bytes in hex of the buffer overflow string will be byte reversed order; Therefore, the four bytes are \x91\x47\x55\x55

2. To calculate how many As are there in the buffer overflow string, we first find out the address of my buffer, where it is 0x7fffffffdc50, and the address of the rbp is 0x7fffffffdc70, the difference between is 32, therefore, 32 + 8 of A needed for the bufferoverflow string.

3. Therefore, the final buffer overflow string is: ‘A’ x 40 “\x91\x47\x55\x55”.

****

# C:\Users\Constantine\Downloads\93511610_2647694092110763_2796084109821083648_n.png

# 

# Problem 7 [18 points]

1. Why is it important for a virus to check if a file has been already infected? **[4 points]**

The reason why it is important for a virus to check if a file has been already infected is because the size of an infected file could grow without bounds through repeated infection.

1. Why are IRC servers chosen by botnet creators for communication between servers and bots? **[3 points]**

The reason is because botnet is more likely to go undetected if the communication between the bots and the C&C server uses standard protocol as opposed to some custom designed protocol, and it is more difficult for a packet sniffer and a protocol analyzer to figure out that anything is awry in a network.

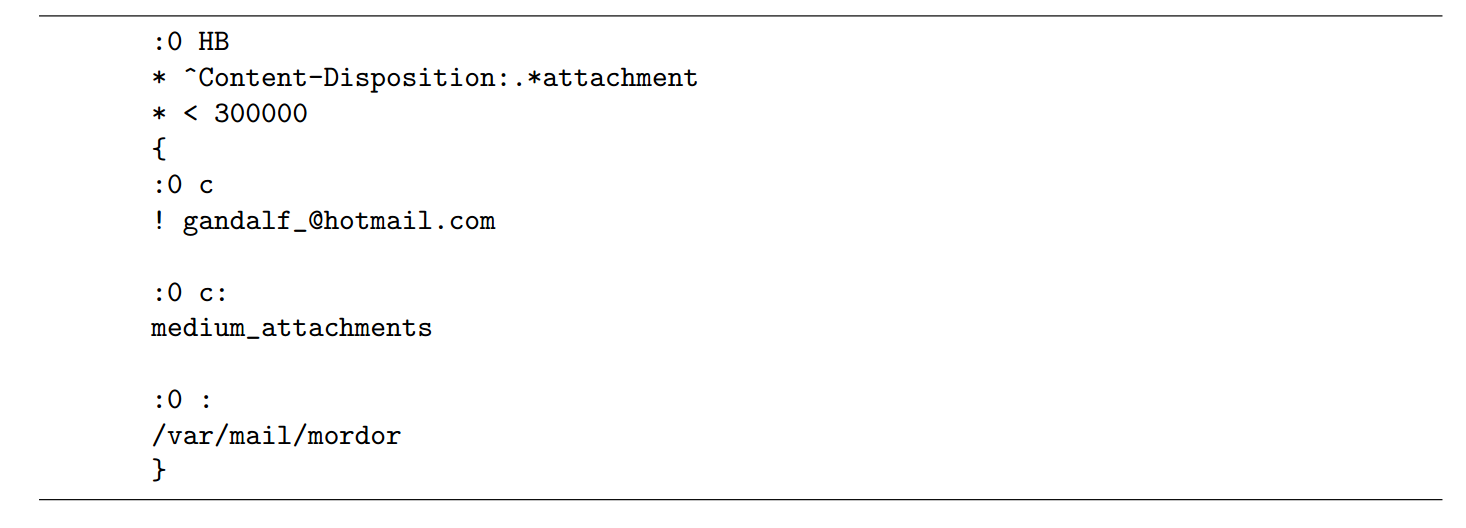
1. What is a content delivery network (CDN)? What makes it secure against DDoS attacks? **[6 points]**
2. A CDN is defined as a network of geographically distributed customer-facing proxy servers that deliver the content in the internet.
3. The main reason why it makes secure against DDoS attacks is that there is a content switch in CDN, that is used for load balancing. This switch would be able to mitigate the attack by sending the incoming traffic to the least loaded server machine.
4. Let’s say an Autonomous System (AS) falls victim to a DDoS attack. In relation to its network neighborhood, explain the actions the AS would take to mitigate this attack.

**[5 points]**

In order to mitigate this attack, the AS can take the following actions:

1. Redirect the traffic to cloud-based traffic scrubbing centers run by DDos mitigation vendors.
2. Implement an instant routing policy change where it can re-route an ongoing DDoS attack as well (because the traffic can be subject to greater filtering).

|  |  |
| --- | --- |
| **Problem 8** | **[9 points]** |
| 1. What does the following spam filter recipe do? | **[5 points]** |



1. The condition < 300000 means that both header and body of the email, the total length of the email should be less than 300000, and the content-Disposition means if the email has the attachment. If an email satisfies the above two conditions, the email will be performing the actions in the brackets as 2. And 3. indicate.
2. The first action will be forward a copy of the email to Gandalf\_@hotmail.com
3. Hold the coming mail if there is one and send the copy of the mail as medium\_attachment.
4. Capture the mail in the /var/mail/mordor

2. Betty Bitdiddle is upset with her friend Ben Bitdiddle because she says he sent her an email message with a virus. This email was in fact not sent by Ben but by an enemy of Betty. If a friend of Betty and Ben had access to all parts of the email, what information could their friend use to clear Ben’s name and show he didn’t actually send it? **[4 points]**

In order to clear Ben’s name, we will have to first check the envelope to find out the servers exchange to validate whether these servers are from the servers normally Ben will use. For example, normally Ben’s email goes through purdue.ecn, however, the spoof one goes through the mit.edu, then you can notice there is a difference.

Second action we can do is to take look at the headers, if the return-path does not match the from email address, then we know there is something wrong with the email.

By checking the header and the MTA, MDA, MUA information, we can clear Ben’s name as described above.

## (If you need extra space, you can use this page)