Basic Web Attacks—Holiday Hack Trail

This lesson teaches that parameters sent from a browser to a web server can be manipulated by a user unless they are protected in some manner. It also shows that attackers can manipulate protected parameters if they can defeat the protection. Web designers, beware! Better yet, do not trust anything that is controlled by the user.

First, watch the 13-minute video by Chris Elgee, the developer of this challenge. He will introduce you to some basic web concepts and a basic attack. Watch carefully for a method you can use to win the game we are about to play. (Note: The Medium and Hard levels of this challenge will require other skills, but this lesson will introduce them.)

Web Apps: a Trailhead

Chris Elgee
Builder & Breaker / Counter Hack Challenges
Twitter @chriselgee

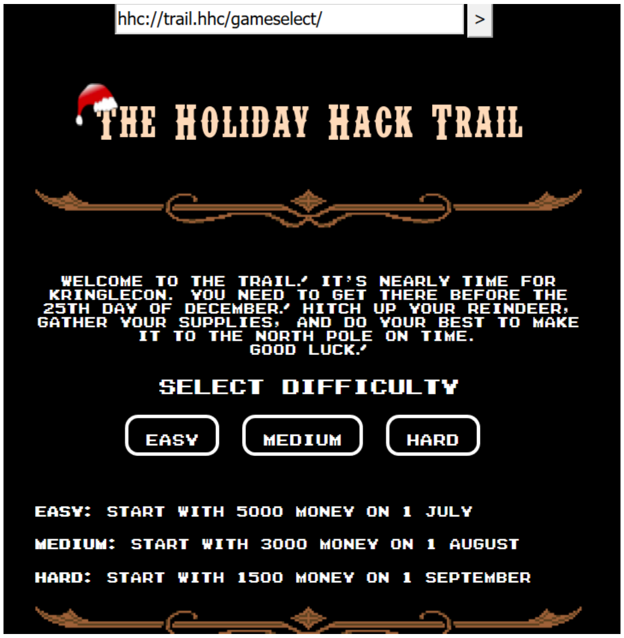
https://www.youtube.com/watch?v=0T6-DQtzCgM&feature=youtu.be

If you are playing the Holiday Hack Challenge (https://holidayhackchallenge.com/2019/) the terminal is located in the Dorm next to the elf Minty Candycane. If you prefer to go directly to the terminal, this link will take you there. https://trail.elfu.org/gameselect/?playerid=JebediahSpringfield

This game is a version of the old Oregon Trail game, which was developed in the early 1970's to teach players about 19th-century pioneer life on the Oregon trail. The graphics represent what was actually available to home computer users in the 1980's. (See http://www.died-of-dysentery.com/stories/early-versions.html Several sites allow you to play the original game online today for free.)

The game we are playing has been specially modified to teach basic vulnerabilities that web developers can program into their sites if they aren't careful. It was developed by Counter Hack Challenges as part of the Holiday Hack Challenge for 2019, Kringlecon2. Kringlecon is a virtual IT security conference that is hosted at the North Pole each year to teach and challenge IT security students and professionals. https://holidayhackchallenge.com/2019/

Here's the beginning of the Trail terminal. The hacks for each game mode, Easy, Medium, and Hard, have the same difficulty as the game modes. After you select your difficulty, the game presents you with the page to purchase supplies. Hah! We're hackers, we don't need supplies.



hhc://trail.hhc/gameselect/

THE HOLIDAY HACK TRAIL

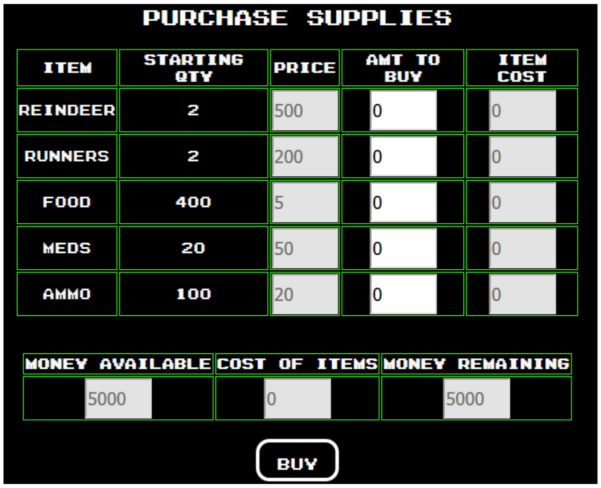
WELCOME TO THE TRAIL! IT'S NEARLY TIME FOR KRINGLECON. YOU NEED TO GET THERE BEFORE THE 25TH DAY OF DECEMBER! HITCH UP YOUR REINDEER, GATHER YOUR SUPPLIES, AND DO YOUR BEST TO MAKE IT TO THE NORTH POLE ON TIME. GOOD LUCK!

SELECT DIFFICULTY

EASY: START WITH 5000 MONEY ON 1 JULY

MEDIUM: START WITH 3000 MONEY ON 1 AUGUST

HARD: START WITH 1500 MONEY ON 1 SEPTEMBER



PURCHASE SUPPLIES

ITEM	STARTING QTY	PRICE	AMT TO BUY	ITEM COST
REINDEER	2	500	0	0
RUNNERS	2	200	0	0
FOOD	400	5	0	0
MEDS	20	50	0	0
AMMO	100	20	0	0

MONEY AVAILABLE	COST OF ITEMS	MONEY REMAINING
5000	0	5000

Easy Mode

Enter the game in Easy mode, and the hacking will be easy too. You can play with supplies on the purchase supplies page or just skip past it by clicking BUY. That will get you started in the game.

Your Task

Cheat! Use the technique Chris describes in the YouTube video to attack the parameter, distance. You should be able to win the game in one turn. In this mode, the game simulates using HTTP GET requests.

Questions

1) How did you change the request so that you won the game?

2) What could the web designer do to prevent you from cheating this way?

3) What was the request method the browser used? (GET, HEAD, POST, CONNECT, and TRACE are possible answers.)

More Fun

Once you've won the game by attacking the distance parameter, you should be able to start over and play the entire game. Except, whenever you need supplies, reindeer, or to improve someone's health you can get what you need by hacking!

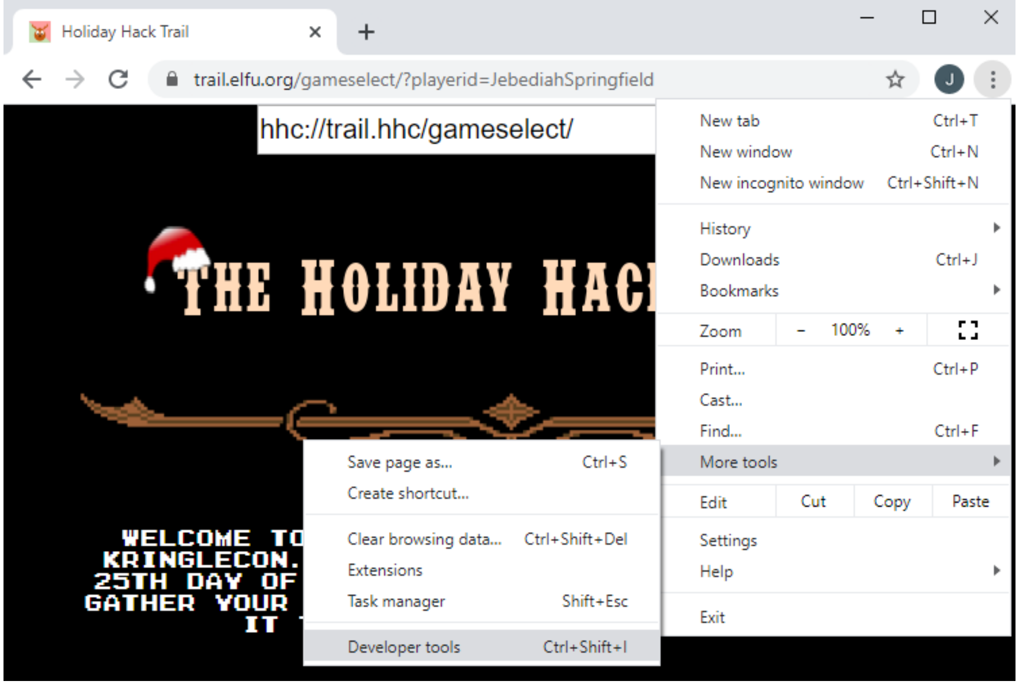
Medium Difficulty Mode

In Medium mode the game submits data to the server using POST requests. Those requests can't be changed as easily as the GET requests can be changed (you just did that in Easy Mode), but that should not deter a hacker.

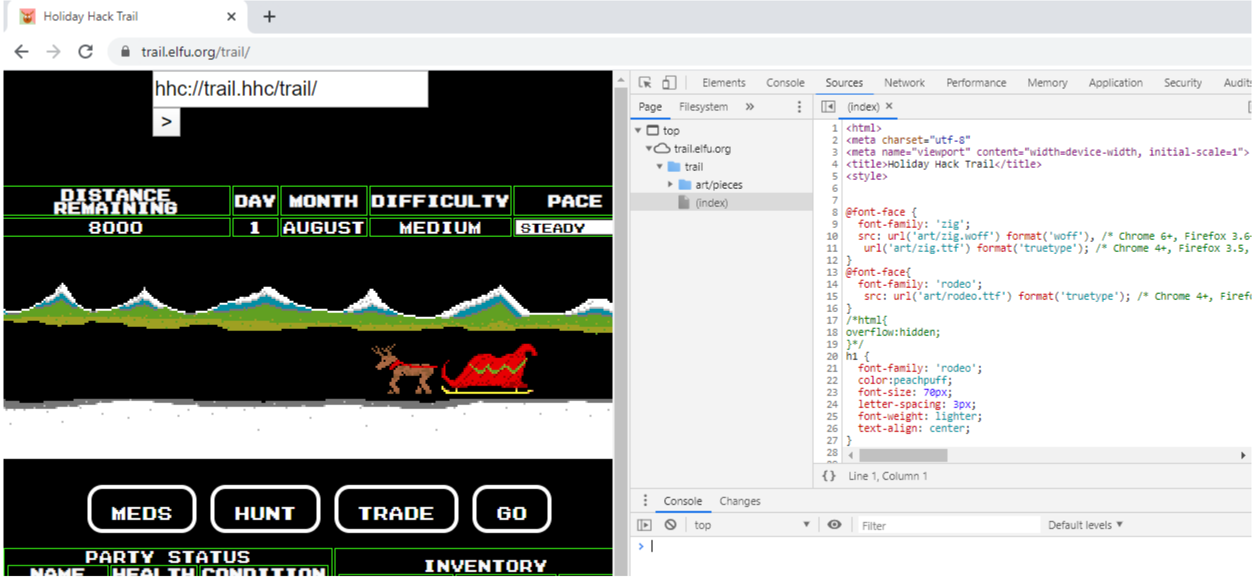
Most modern browsers include developer tools, which can be used to examine the contents of a web page in detail. The lesson will use Google Chrome, but most other browsers have similar capability.

Step 1: A quick tour of Chrome Developer tools

Go to the trail URL, https://trail.elfu.org/gameselect/?playerid=JebediahSpringfield, and open Developer tools. `Ctrl-Shift-I` works if you like keyboard shortcuts.



If you select the Sources tab, and highlight (index) you should see the HTML source code for the page.

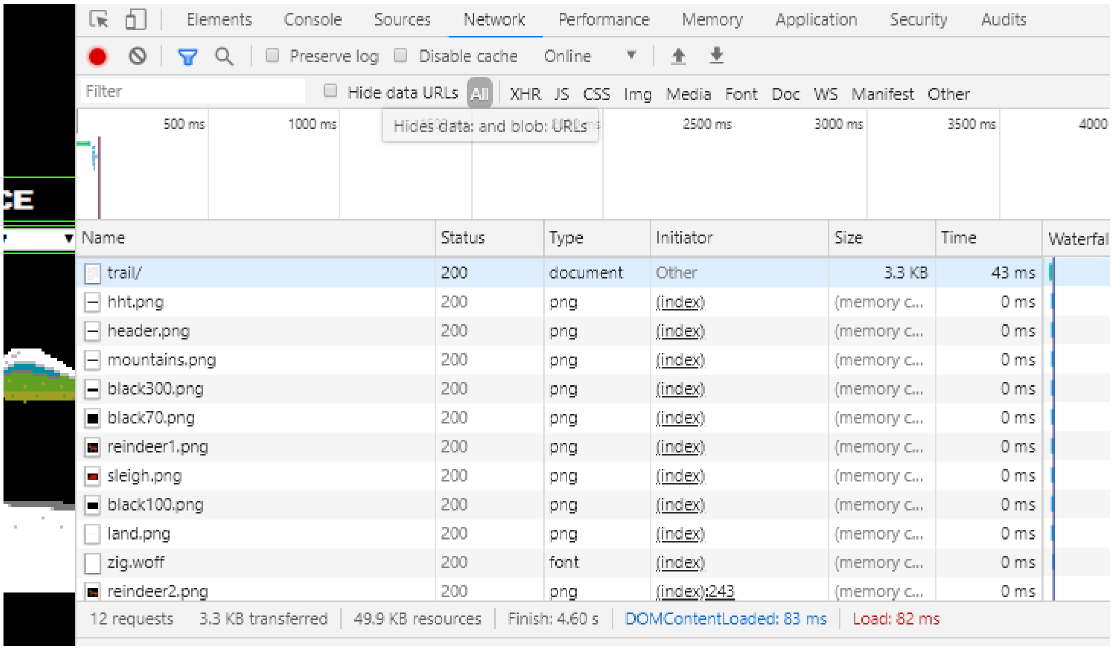


```
<html>
<meta charset="utf-8"
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Holiday Hack Trail</title>
<style>


@font-face {
  font-family: 'zig';
  src: url('art/zig.woff') format('woff'), /* Chrome 6+, Firefox 3.6
   url('art/zig.ttf') format('truetype'); /* Chrome 4+, Firefox 3.5,
}
@font-face{
  font-family: 'rodeo';
   src: url('art/rodeo.ttf') format('truetype'); /* Chrome 4+, Firef
}
/*html{
overflow:hidden;
}*/
h1 {
  font-family: 'rodeo';
  color:peachpuff;
  font-size: 70px;
  letter-spacing: 3px;
  font-weight: lighter;
  text-align: center;
}
```

The index.html file was downloaded from the server; it is the original code for the page. We'll get to this in a minute, but JavaScript in the page can modify the original code. Since this is the original code, we can't change it in Dev Tools.

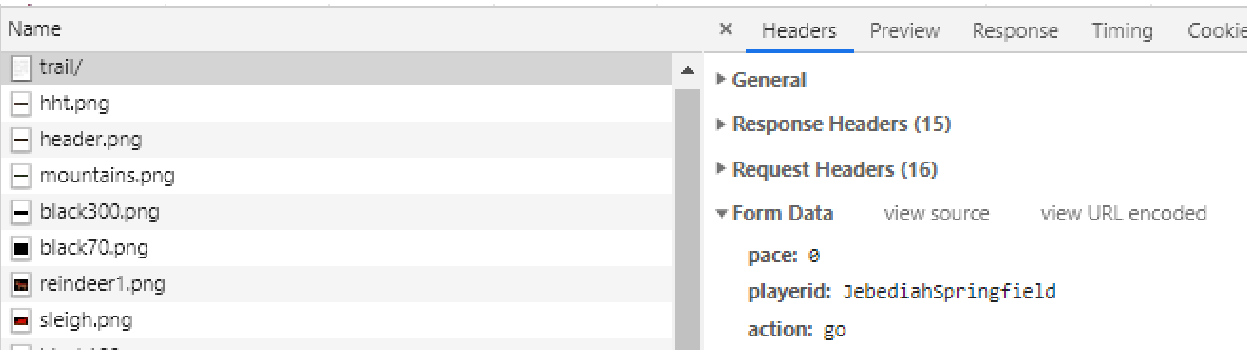
Switch to the Network tab, and then click GO on the web site once or twice to generate traffic.



Name	Status	Type	Initiator	Size	Time	Waterfall
trail/	200	document	Other	3.3 KB	43 ms	
hht.png	200	png	(index)	(memory c...	0 ms	
header.png	200	png	(index)	(memory c...	0 ms	
mountains.png	200	png	(index)	(memory c...	0 ms	
black300.png	200	png	(index)	(memory c...	0 ms	
black70.png	200	png	(index)	(memory c...	0 ms	
reindeer1.png	200	png	(index)	(memory c...	0 ms	
sleigh.png	200	png	(index)	(memory c...	0 ms	
black100.png	200	png	(index)	(memory c...	0 ms	
land.png	200	png	(index)	(memory c...	0 ms	
zig.woff	200	font	(index)	(memory c...	0 ms	
reindeer2.png	200	png	(index):243	(memory c...	0 ms	

12 requests | 3.3 KB transferred | 49.9 KB resources | Finish: 4.60 s | DOMContentLoaded: 83 ms | Load: 82 ms

You can see all the network traffic between your browser and the server. You can see that the browser downloaded several image files (.png) from the server. Find `trail/` in your list and click on it. It contains the code for the section of the web site we are are using now. (The address for the page we are looking at is trail.elfu.org/trail/, or trail/ for short.)



pace: 0
playerid: JebediahSpringfield
action: go

Open General at the right and answer these questions.

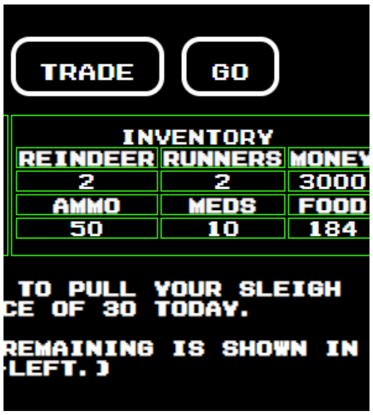
Questions

4) What was the request method the browser used?

5) What status code did the server return?

6) What is the IP address of the web server?

Look at the Form Data on the right side. It should be close to the information showing on the display of the web site (it tells the browser what the next values will be, so it is off by a little.) If the web site says you have two reindeer, the Form Data should too.

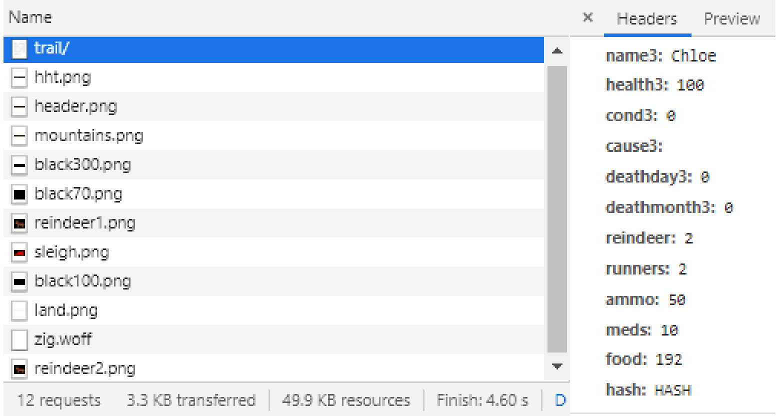


INVENTORY

REINDEER	RUNNERS	MONEY
2	2	3000
AMMO	MEDS	FOOD
50	10	184

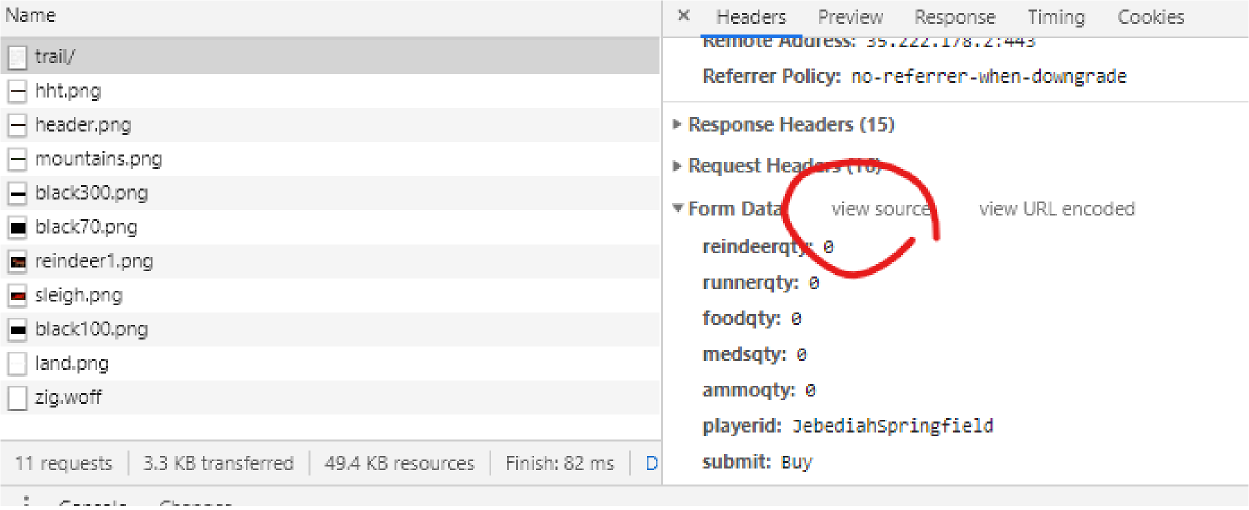
TO PULL YOUR SLEIGH
CE OF 30 TODAY.

REMAINING IS SHOWN IN
LEFT.)



name3: Chloe
health3: 100
cond3: 0
cause3:
deathday3: 0
deathmonth3: 0
reindeer: 2
runners: 2
ammo: 50
meds: 10
food: 192
hash: HASH

Now look for the View Source legend at the top of the Form Data and click it.

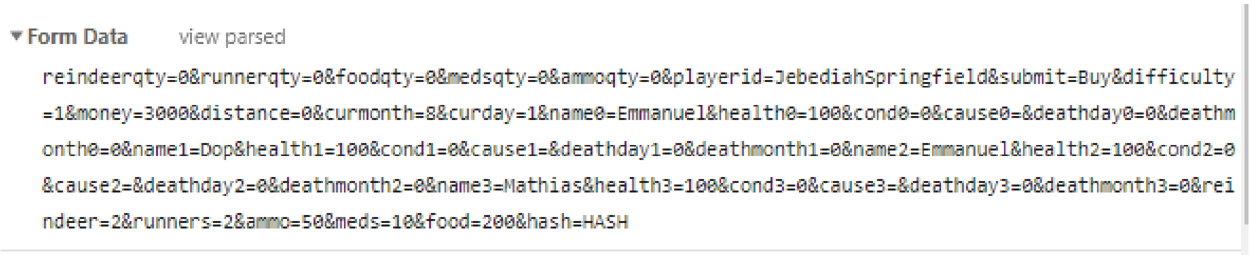


```
Name
trail/
hht.png
header.png
mountains.png
black300.png
black70.png
reindeer1.png
sleigh.png
black100.png
land.png
zig.woff

11 requests | 3.3 KB transferred | 49.4 KB resources | Finish: 82 ms
```

```
Remote Address: 35.222.178.2:443
Referrer Policy: no-referrer-when-downgrade
▸ Response Headers (15)
▸ Request Headers (16)
▾ Form Data    view source    view URL encoded
   reindeerqty: 0
   runnerqty: 0
   foodqty: 0
   medsqty: 0
   ammoqty: 0
   playerid: JebediahSpringfield
   submit: Buy
```

This shows you what the form data looks like when it is sent to the server.



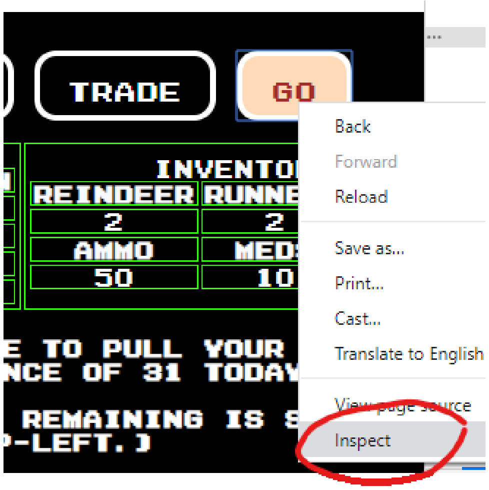
```
reindeerqty=0&runnerqty=0&foodqty=0&medsqty=0&ammoqty=0&playerid=JebediahSpringfield&submit=Buy&difficulty
=1&money=3000&distance=0&curmonth=8&curday=1&name0=Emmanuel&health0=100&cond0=0&cause0=&deathday0=0&deathm
onth0=0&name1=Dop&health1=100&cond1=0&cause1=&deathday1=0&deathmonth1=0&name2=Emmanuel&health2=100&cond2=0
&cause2=&deathday2=0&deathmonth2=0&name3=Mathias&health3=100&cond3=0&cause3=&deathday3=0&deathmonth3=0&rei
ndeer=2&runners=2&ammo=50&meds=10&food=200&hash=HASH
```

Step 2 – Change the page in Developer Tools

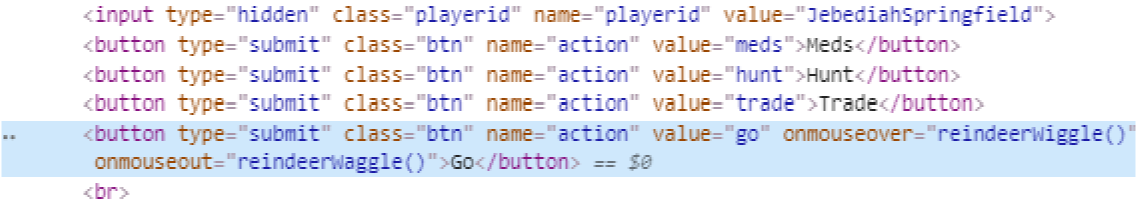
The source code, index.html or whatever, is the code that was originally delivered to the browser. Since JavaScript can modify the code on the fly, the browser keeps a copy of the current version in the DOM tree. In Chrome Dev Tools, the DOM tree is shown under the Elements tab. We can change things here, since this is the current version of the code. For more information on the Elements tab, read https://developers.google.com/web/tools/chrome-devtools/dom. To see more about the Document Object Model (DOM) in general, read https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

Most of the tree is closed when we start; for a modern web site the tree can be quite long. Rather than open every element until we find what we want, there are two ways to find what we want quickly.

1. On the web page itself, right-click something of interest and select Inspect. GO was clicked in this example.

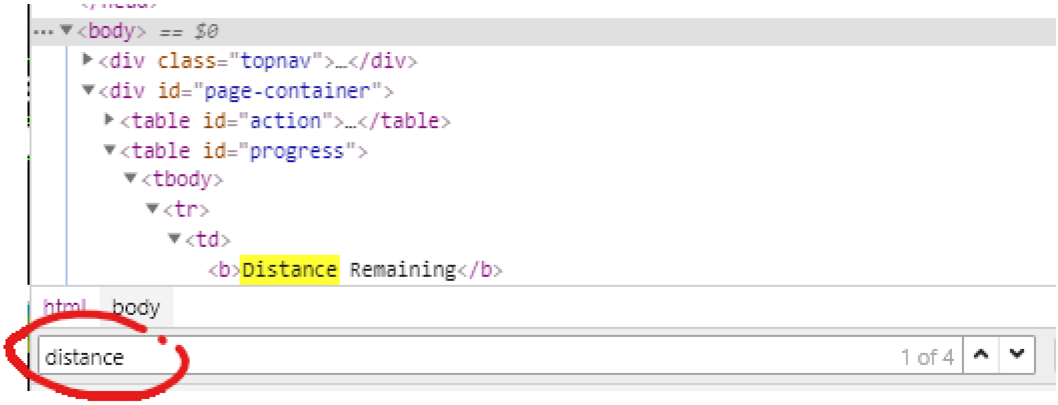


You will be taken to that part of the code.



```
<input type="hidden" class="playerid" name="playerid" value="JebediahSpringfield">
<button type="submit" class="btn" name="action" value="meds">Meds</button>
<button type="submit" class="btn" name="action" value="hunt">Hunt</button>
<button type="submit" class="btn" name="action" value="trade">Trade</button>
<button type="submit" class="btn" name="action" value="go" onmouseover="reindeerWiggle()"
 onmouseout="reindeerWaggle()">Go</button> == $0
<br>
```

2. If you are looking for a name, click in the Elements column, type Ctrl-F (find), and then enter the name. Type the name, distance for example, in the box and then cycle though all the times it occurs in the code.



```
... ▼<body> == $0
  ▶<div class="topnav">…</div>
  ▼<div id="page-container">
    ▶<table id="action">…</table>
    ▼<table id="progress">
      ▼<tbody>
        ▼<tr>
          ▼<td>
             <b>Distance Remaining</b>
```

You should be able to find the “distance” in the portion of the code that sends all the values back to the server. Change it and win the game!

Questions

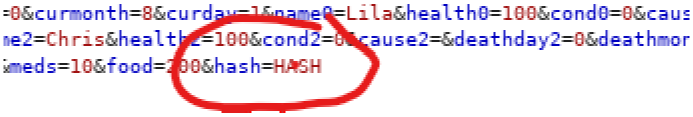
7) What did you do to win the game?

More Fun

Now that you know the secret of modifying requests (answer to question 6), you have the game at your mercy. You can play the game and give yourself reindeer, supplies, and health any time you need it.

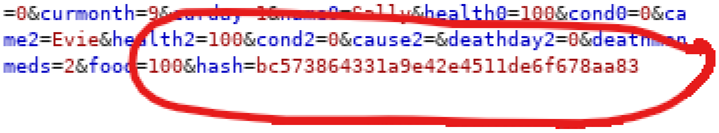
HARD Mode

It really is harder to hack the game in Hard mode. You may not have noticed it, but the request in Medium mode had `hash=HASH` at the end of the data.



```
=0&curmonth=8&curday=1&name0=Lila&health0=100&cond0=0&caus
me2=Chris&health2=100&cond2=0&cause2=&deathday2=0&deathmor
&meds=10&food=200&hash=HASH
```

It wasn't doing anything in Medium mode. In Hard mode it looks like this.



```
=0&curmonth=9&curday=1&name0=Sally&health0=100&cond0=0&ca
me2=Evie&health2=100&cond2=0&cause2=&deathday2=0&deathmon
meds=2&food=100&hash=bc573864331a9e42e4511de6f678aa83
```

That hash is protecting the data to keep evil people (us) from tampering with the data in the POST request.

What is a Hash?

The hash function takes data as input and hashes or grinds it up until the output is totally unrecognizable. A cryptographic hash has three main properties:

1) If you hash the same data again, the output hash will be the same. If even one bit of the input is different, however, the output hash will be completely different. This is very useful if you wish to detect if the data has been changed.
2) It should be very difficult* to alter the data and still get the same hash output. For some older hash functions, it is possible to tamper with the data and get the same hash output, but those hash functions should not be used in current software.
3) It should be very difficult* to compute what the input was when someone gives you the hash output.

Note: "Very difficult" means it should take many years on a huge computer to do this.

For more information about cryptographic hashes, you can start here.
https://en.wikipedia.org/wiki/Cryptographic_hash_function

Crack the Hash

If we want to defeat the hash that protects the data, we need to know what input generated the hash we see in the image above. "Wait!" I hear you say, "You just said you cannot compute what the input was when you are given a hash output!" That is true, we cannot. What we can do is to find many possible values for what the input could be, compute the hash for each one, and then check to see if any of the answers match the hash we have. This works well for hashes of people's passwords, since most people choose poor passwords.

First, we need to know what algorithm was used to create this hash. If you count the characters in the image above, you should get 32 as your answer. It takes two characters to fill one byte of data, so the hash is 32/2 = 16 bytes long. There are 8 bits in a byte, so the hash is 16*8 = 128 bits long. Use your search engine skill to determine the commonly used hash function that creates an output that is 128 bits long.

Question

8) What is, most likely, the hash function that created the hash in the image above?

You can use either hashcat or JohnTheRipper to crack the hash for the image above. However, there is an easier way for simple hashes: a search engine. Google worked best for me on this one. Search for `{answer to question 8 goes here, no braces} hash "bc573864331a9e42e4511de6f678aa83"`

The string in quotes is the hash from the image above. The quotes require Google to only show answers that include that string.

Question

9) What was the input to the hash function, so that "bc573864331a9e42e4511de6f678aa83" was the output?

If you didn't come across it in your search (it was 3rd in my search,) this site is very handy for what we are doing. http://hash.oderskebrzdy.cz/md5.php?kolik=0-100000 (We could crack the hash each time we needed to, tamper with our data, and then compute a new hash using md5sum, but why bother?) Note: you could easily make your own table like this with just a few lines of code.

Begin your attack

Use Chrome Dev Tools to modify what the browser sends to the Holiday Hack Trail site just like we did for the Medium challenge. The difference is that you will have to determine how the hash changes when you tamper with a parameter. There are a lot of parameters in the data, but to win the game the only parameter you need is distance. Just worry about distance for now.

- Click GO and look at the data sent to the server.
- Crack the hash (look it up on the site above)
- Click GO and look at the data again. What has changed since the last time?
- Crack the hash again and see how it changed.
- Repeat until you have determined what a change in the distance does to the number that gets hashed.

Hint: The relationship between the change in distance and the change in the number that gets hashed is very simple. Also, the talk that started this lesson has a portion of the server code in one of the slides; it is the one where he talks about writing the site in Python.

Once you know how the hash changes when you tamper with distance, you can win the game.

- Find the parts you want to change in the Dev Tools Elements tab.
- Record the current distance
- Change the distance in the request to the value that wins the game

- Compute how much you changed the distance
- Crack the hash using the website
- Use the relationship you discovered above and compute the number you need to hash
- Get the number’s hash from the website
- Change the hash in the request
- Click Forward and win the game.

Go Forth and Conquer!

Questions

10) What is the relationship between the change in distance and the change in number that is hashed?

11) How did you beat the game?

12) Can you think of a way that this vulnerability could be fixed? Remember that any code on the browser can be studied or changed by the attacker.

Optional—Burp Suite

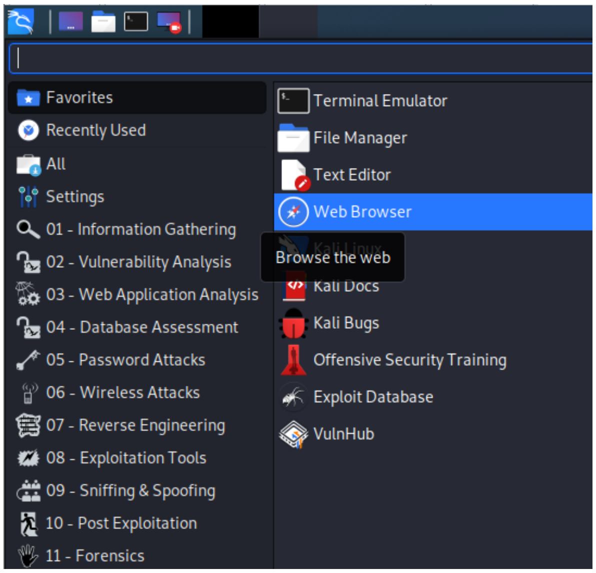
Burp Suite is a powerful tool that allows people testing Web App security it intercept traffic the browser sends to the server, and manipulate it. It is the same thing that happens in a Man in the Middle (MitM) attack, where an attacker intercepts your communications. Your teacher can use the information in the teacher notes section to prepare it for you. If you are working on your own, you can download the Community Edition for either Windows or Linux. Another option is to download Kali Linux, which comes with Burp Suite installed.

These instructions will assume that you are using the Burp Suite installed on Kali Linux (a virtual machine is fine.) The browser that Kali uses is Firefox. However, different versions of Kali use different versions of Firefox, so the settings you need to make may be in slightly different locations. These instructions are for Firefox v68 which comes with Kali 2019.4.

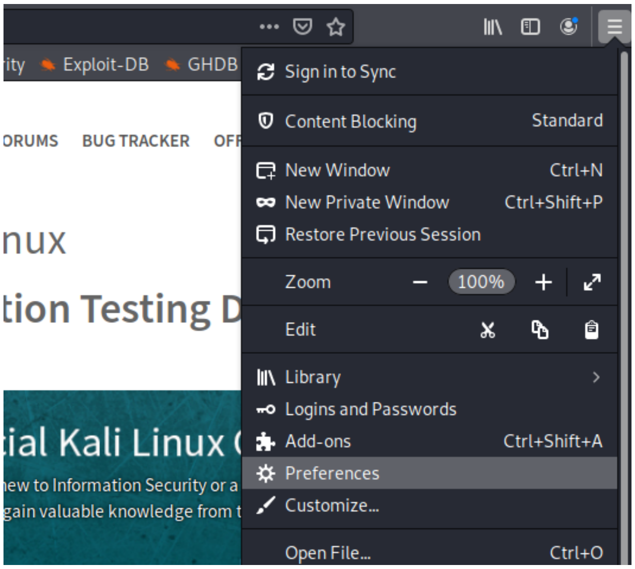
Note: If your instructor has configured your environment ahead of time (Virginia Cyber Range) you may be able to skip ahead to step 3.

Step 1—Configure the browser to use Burp Suite

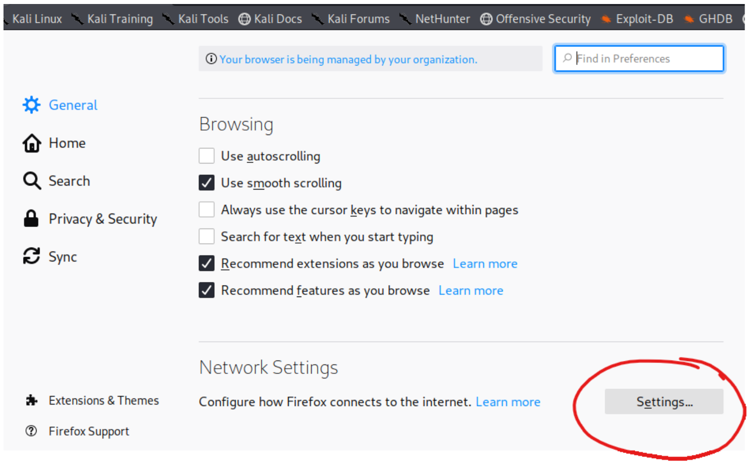
In Kali you need to open your web browser.



Change the browser proxy settings to point to 127.0.0.1 8080, which is the Burp default. A proxy server intercepts the traffic to and from your browser. Schools usually use proxy servers to meet requirements for protecting students against pornography and other evil. Corporations use proxies to enforce security policy and provide some protection against attacks. We will use Burp as a proxy server to hack the Holiday Trail game. In the browser, select Preferences.



Scroll to the bottom and select Settings.



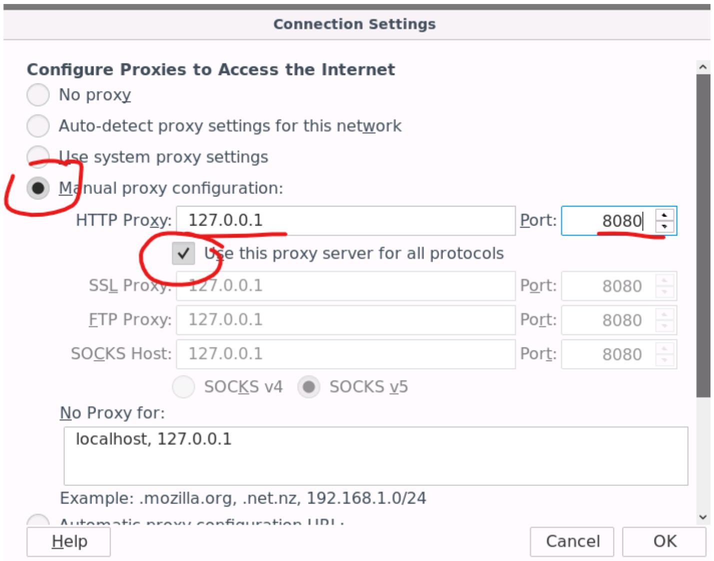
Browsing

- ☐ Use autoscrolling
- ☑ Use smooth scrolling
- ☐ Always use the cursor keys to navigate within pages
- ☐ Search for text when you start typing
- ☑ Recommend extensions as you browse Learn more
- ☑ Recommend features as you browse Learn more

Network Settings

Configure how Firefox connects to the internet. Learn more [Settings...]

Select Manual proxy configuration, Use this proxy server for all protocols, and enter 127.0.0.1 and port 8080.



Connection Settings

Configure Proxies to Access the Internet

- ○ No proxy
- ○ Auto-detect proxy settings for this network
- ○ Use system proxy settings
- ● Manual proxy configuration:

HTTP Proxy: 127.0.0.1 Port: 8080

☑ Use this proxy server for all protocols

SSL Proxy: 127.0.0.1 Port: 8080

FTP Proxy: 127.0.0.1 Port: 8080

SOCKS Host: 127.0.0.1 Port: 8080

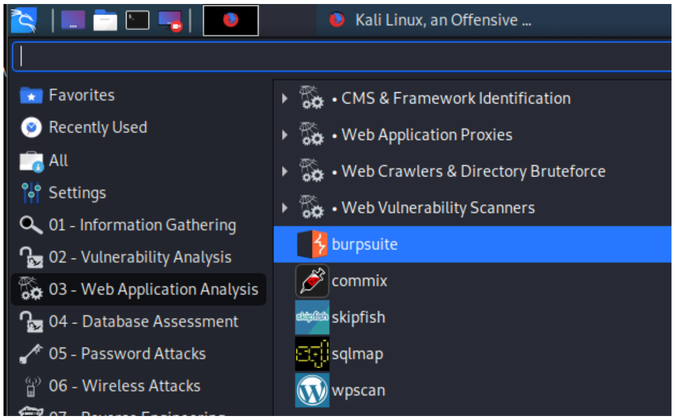
○ SOCKS v4 ● SOCKS v5

No Proxy for:

localhost, 127.0.0.1

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Now we have to start Burp. Your browser won't work until Burp is running since we just configured it to send its data to Burp instead of the Internet.



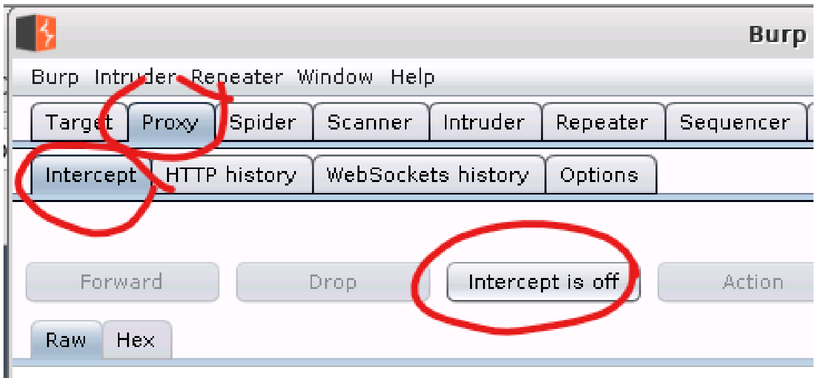
Kali Linux, an Offensive ...
Favorites
Recently Used
All
Settings
01 - Information Gathering
02 - Vulnerability Analysis
03 - Web Application Analysis
04 - Database Assessment
05 - Password Attacks
06 - Wireless Attacks
07 - Reverse Engineering
• CMS & Framework Identification
• Web Application Proxies
• Web Crawlers & Directory Bruteforce
• Web Vulnerability Scanners
burpsuite
commix
skipfish
sqlmap
wpscan

Burp Suite will show you a license agreement. If it complains about the Java version, accept the message and continue. Don't allow Burp Suite to update itself. Since Next is the only choice for projects in the free community edition, click next. Then click Start Burp and you should be ready to go.

We also have to tell our browser to accept Burp's digital certificate. Burp will be fooling our browser, and will present it's own TLS certificates in place of the real ones for the sites we visit. If we don't tell the browser to believe everything that Burp tells it, the browser will generate errors or block the sites. Follow the instructions here. https://support.portswigger.net/customer/portal/articles/1783075-installing-burp-s-ca-certificate-in-your-browser The browser in Kali Linux is a variant of Firefox, so use those instructions. Note: if http://burp gives an error instead of the screen shown in the article, use http://127.0.0.1:8080 instead. Burp has to be running for this to work.

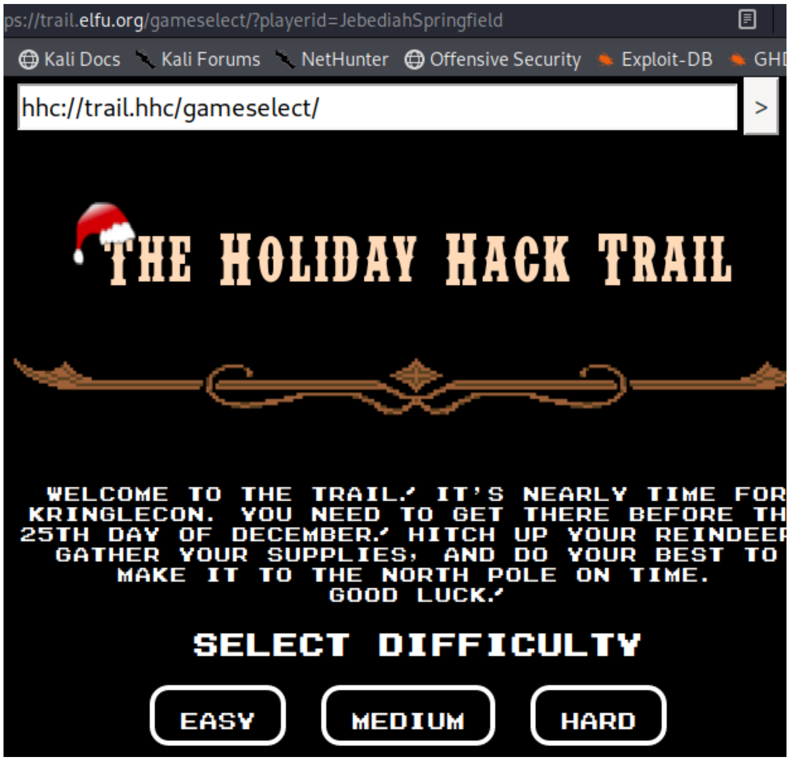
Step 2. Set Burp Proxy to Intercept Off

While we are getting started, we want Burp to allow requests from our browser to go straight through to the web. We'll use Intercept in a bit, but for now set Proxy, Intercept, Intercept Off.



Step 3. Browse to https://trail.elfu.org

If the proceeding steps were done correctly, you should be able to browse to https://trail.elfu.org.

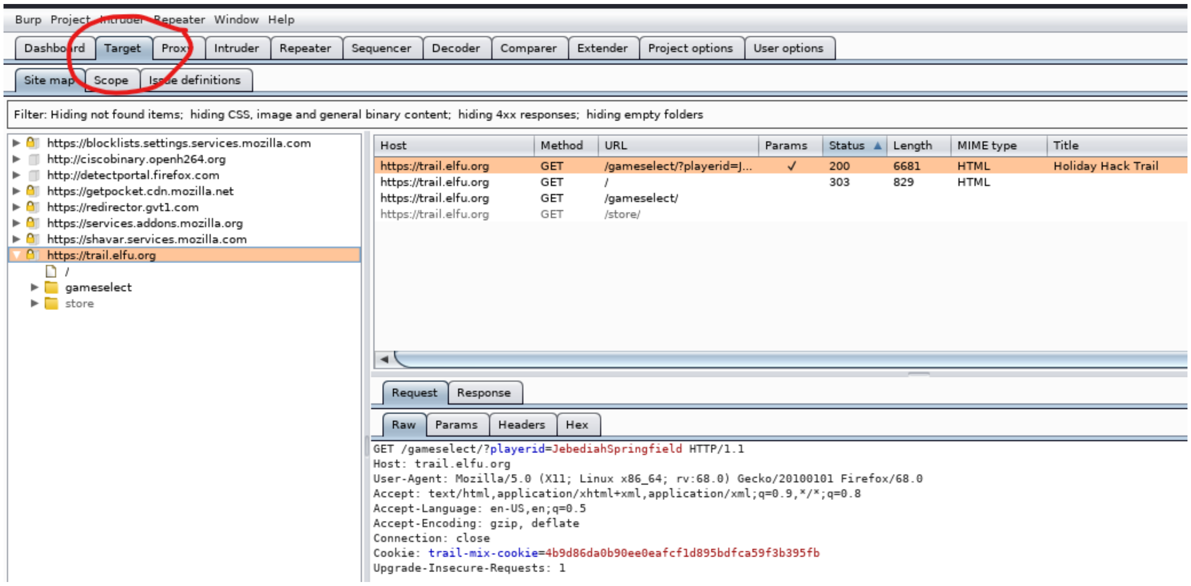

ps://trail.elfu.org/gameselect/?playerid=JebediahSpringfield
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHI
hhc://trail.hhc/gameselect/
>

THE HOLIDAY HACK TRAIL

WELCOME TO THE TRAIL! IT'S NEARLY TIME FOR
KRINGLECON. YOU NEED TO GET THERE BEFORE TH
25TH DAY OF DECEMBER! HITCH UP YOUR REINDEEF
GATHER YOUR SUPPLIES, AND DO YOUR BEST TO
MAKE IT TO THE NORTH POLE ON TIME.
GOOD LUCK!

SELECT DIFFICULTY

When you go to the Target Tab of Burp Suite, you should see the traffic between the browser and the server, as well as the content of each session. There will be sites besides trail.elfu.org, even if you didn't browse to anything else. Browsers are chatty. You can get an idea of what HTTP traffic looks like just by playing in the target section.



Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

```
https://blocklists.settings.services.mozilla.com
http://ciscobinary.openh264.org
http://detectportal.firefox.com
https://getpocket.cdn.mozilla.net
https://redirector.gvt1.com
https://services.addons.mozilla.org
https://shavar.services.mozilla.com
https://trail.elfu.org
    /
    gameselect
    store
```

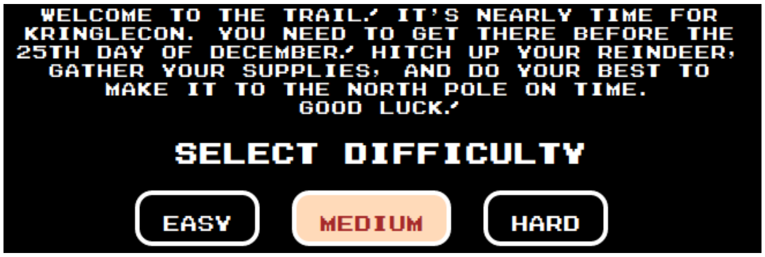
Host	Method	URL	Params	Status	Length	MIME type	Title
https://trail.elfu.org	GET	/gameselect/?playerid=J...	✓	200	6681	HTML	Holiday Hack Trail
https://trail.elfu.org	GET	/		303	829	HTML	
https://trail.elfu.org	GET	/gameselect/					
https://trail.elfu.org	GET	/store/					

```
GET /gameselect/?playerid=JebediahSpringfield HTTP/1.1
Host: trail.elfu.org
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: trail-mix-cookie=4b9d86da0b90ee0eafcf1d895bdfca59f3b395fb
Upgrade-Insecure-Requests: 1
```

Note: If you don't see anything at all when you attempt to browse, it is possible you forgot to select Intercept Off on the previous page.

Step 4: Prepare to attack

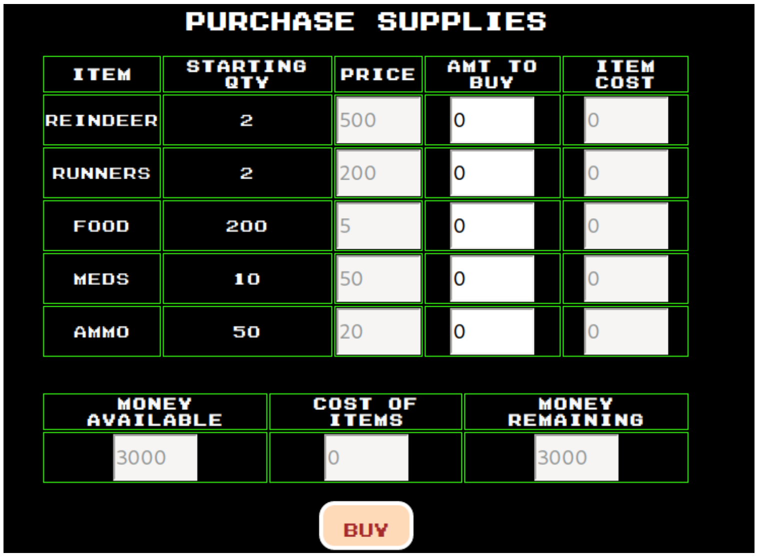
Select the MEDIUM difficulty for Holiday Hack Trail.



WELCOME TO THE TRAIL! IT'S NEARLY TIME FOR KRINGLECON. YOU NEED TO GET THERE BEFORE THE 25TH DAY OF DECEMBER! HITCH UP YOUR REINDEER, GATHER YOUR SUPPLIES, AND DO YOUR BEST TO MAKE IT TO THE NORTH POLE ON TIME. GOOD LUCK!

SELECT DIFFICULTY

Skip past the Purchase Supplies page by selecting BUY. We are hacking, we won't need supplies.



PURCHASE SUPPLIES

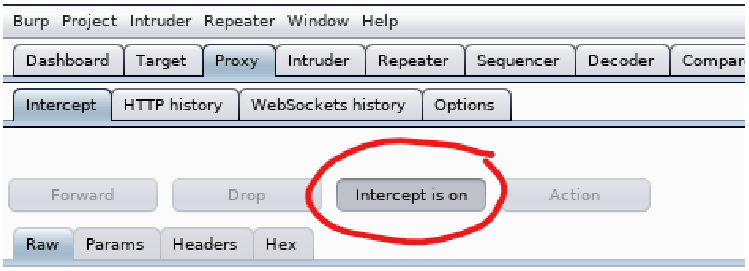
ITEM	STARTING QTY	PRICE	AMT TO BUY	ITEM COST
REINDEER	2	500	0	0
RUNNERS	2	200	0	0
FOOD	200	5	0	0
MEDS	10	50	0	0
AMMO	50	20	0	0

MONEY AVAILABLE	COST OF ITEMS	MONEY REMAINING
3000	0	3000

When you start the game proper, you are ready to attack.

5. Attack

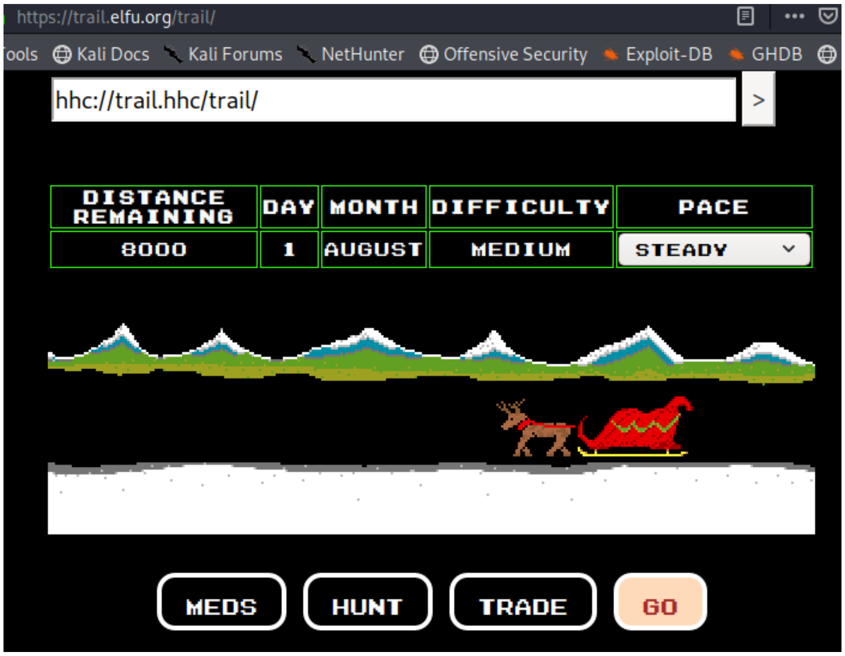
Now we want Burp to Intercept each transaction between the browser and the Holiday Hack Trail web site so that we can adjust things to our liking. In Burp, go to the Proxy tab, Intercept, and turn Intercept On.



Now, any time the browser sends traffic, Burp will stop it to allow you to inspect or modify it. When you are satisfied, click Forward and Burp will release the traffic. Until you click Forward, the browser won't show anything—it will look like the network has gone down.

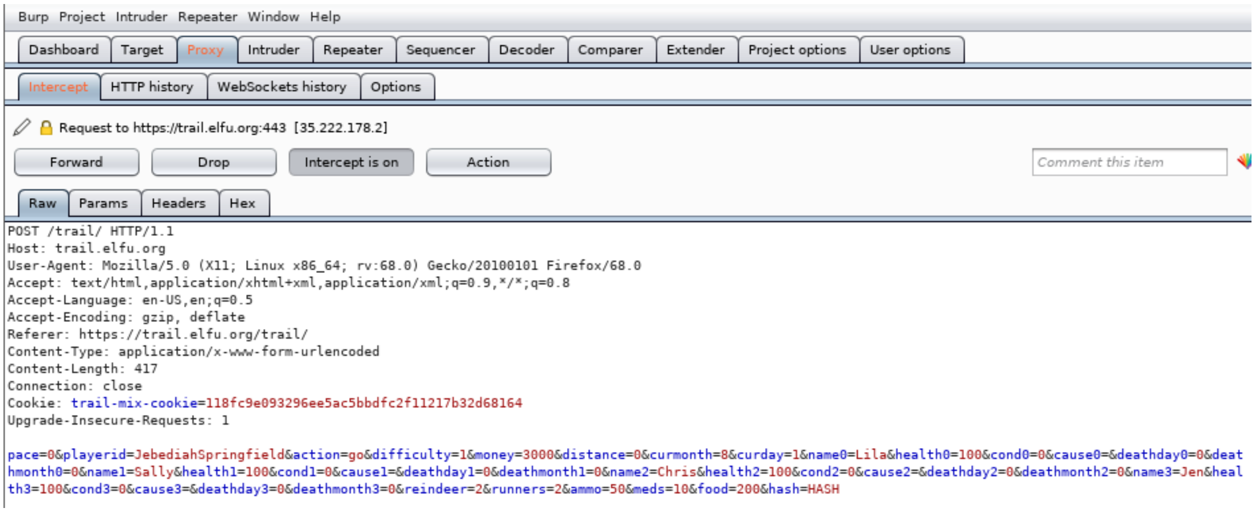
Note: The browser is chatty and talks to mozilla.org and similar sites without your telling it to. You may have to click Forward to get those out of your way.

On the web browser at the Holiday Hack Trail site, click GO to get started.



DISTANCE REMAINING	DAY	MONTH	DIFFICULTY	PACE
8000	1	AUGUST	MEDIUM	STEADY

Go back to Burp, and you should see something like this.



Request to https://trail.elfu.org:443 [35.222.178.2]

```
POST /trail/ HTTP/1.1
Host: trail.elfu.org
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://trail.elfu.org/trail/
Content-Type: application/x-www-form-urlencoded
Content-Length: 417
Connection: close
Cookie: trail-mix-cookie=118fc9e093296ee5ac5bbdfc2f11217b32d68164
Upgrade-Insecure-Requests: 1

pace=0&playerid=JebediahSpringfield&action=go&difficulty=1&money=3000&distance=0&curmonth=8&curday=1&name0=Lila&health0=100&cond0=0&cause0=&deathday0=0&deat
hmonth0=0&name1=Sally&health1=100&cond1=0&cause1=&deathday1=0&deathmonth1=0&name2=Chris&health2=100&cond2=0&cause2=&deathday2=0&deathmonth2=0&name3=Jen&heal
th3=100&cond3=0&cause3=&deathday3=0&deathmonth3=0&reindeer=2&runners=2&ammo=50&meds=10&food=200&hash=HASH
```

If you see something different, it may be the browser chatting with Mozilla. Click Forward and maybe you'll get to the picture above. If not, go back to Holiday Hack Trail and click GO to try again.

You can edit the parameters the browser is sending to the site as you choose, within reason. When you have finished your edits, click Forward to send them to the site.

Now you have everything you need to win the game in one move; it is right there in front of you!

Hard Mode

The same technique works in hard mode, except that you must adjust the hash as well.