Arthur Viegas Eguia

S - Spoofing

If tapirs prove to be popular, and if the website starts getting a lot of traffic and making money, people might start scamming with websites similar to tapirsunlimited.com. They might create websites like TapirsUnlimited.com (that's an uppercase Tau), or tapirsunlimitedd.com. Both are used to steal people's accounts and/or get their money. People are unlikely to type the first, but they might click it if it is in a malicious email. The second is the product of a typo and is likely to happen.

This attack is not that hard to conduct, it does not target the Tapirs Unlimited system per se. Scammers can create a website that looks exactly like Tapirs Unlimited, or at least have a User Interface that is similar enough. They don't need to model every single page, just the ones they intend to use for their scam, probably those related to login, password management, and payment information. This attack demands knowledge in programming to copy the user interface with a sufficient level of detail. The hardest part would be the social engineering: coming up with a plausible explanation as to why people should access these websites and type their personal information. This can be done through emails, for example.

Mitigation: this one is a bit hard to mitigate. Scammers are very talented at copying websites (even copying the design fairly well) and creating plausible-ish stories for their social engineering. What is more, some people do not expect to be scammed and believe those stories/fake websites.

However, there are some guidelines that can be followed to make Tapirs Unlimited more reliable. The first is a having valid certificate that can be easily verified by the user's browser (not all scam websites have those). What is more, they need an email domain that will send all official communications and can be easily linked to the website (something like @tapirsunlimited.com). If the official communication is sent from a @gmail.com or @outlook.com email address it is hard to tell official communication apart from scams. Therefore, creating a TapirsUnlimited email domain is a must. All the communications must be typo-free and use good grammar (which scamming emails usually lack). It is also important that the website has a very professional, well-designed, reliable user interface which sets it apart from fake poorly designed websites. If one looks at scammer websites they often have a clunky design or do not follow consistent design patterns.

If the company has enough money, it can register a large number of domains that can be the result of a typo (TTapirsunlimited, taapirsunlimited, yapirsunlimited, tapirsilimited, tairsunliited). This way many common domain name typos will point to the real Tapirs Unlimited website. It is hard to verify and register all possible domains which are the result of a typo. So they can have a team that looks for approximations and domains that can be typed by mistake. Last but not least, said team can write a script on a bunch of variations of the domain name to see if some of them lead to a scam website. The company can then report them as fraud, getting them offline.

This is an example of Spoofing, as someone is pretending to impersonate someone they are not. In other words, a scammer is pretending to be Tapirs Unlimited.

T - Tampering

One example of how this could be done is through an adversary-in-the-middle attack. If the site does not have a certificate, a malicious user can be in the middle of a legitimate user and the server. Let the user be Alice, Tapirs Unlimited be Bob and the adversary-in-the-middle be Mal.

Alice thinks she is talking to Bob. However, she is talking to Mal. Similarly, Bob thinks he is talking to Alice, but he is talking to Mal. When Alice sends a request to connect, Mal interceptes it, pretends to be Bob, and initiates a connection with her. They will then do the Diffie-Hellman and establish a Symmetric Encryption key among them. Mal, who was pretending to be Alice, then connects to Bob, and they create a different Symmetric Encryption Key via Diffie-Hellman. There are two connections, one between Alice and Mal and another between Mal and Bob. Alice sends Bob messages. Mal intercepts them, reads them, and potentially modifies them. If Alice makes a purchase, Mal might change the amount Alice pays, her address, or the amount of each item purchased. Mal can read and modify the packages freely. He can also modify her posts, among other actions.

If the connection is using HTTP, instead of HTTPS, then the Diffie-Hellman key exchange does not happen and the packets are sent in plain text. Mal will just intercept the packages Alice sends and modify them before sending them along to Bob.

The best way to avoid an adversary in the middle attack is to get a certificate and use HTTPS exclusively. The certificate encrypts the correct public key for Bob and is signed by a mutually trusted certificate authority. Alice will validate the certificate using the certificate authority's public key and pose a challenge to Bob. She will send a random number to be encrypted with his private key. He will send back the random number with the B he is using for the Diffie-Hellman key, this message will be encrypted using his private key (they might also be hashed). Alice will decrypt the message using Bob's public key. If the number was correctly encrypted and if the Diffie-Hellman number checks up then there are no adversaries in the middle. After going through all these steps they can communicate with each other without the threat of an Adversary in the Middle.

This is a classic example of tampering as there is a malicious user reading and modifying the data sent over the network. They are tampering with Alice's input.

R - Repudiation

One way to do this is through an SQL injection, in the login form (before the user is identified). This way the system does not tag the action to a specific account. The attacker can append after the fake username and password a sting like "; DROP DATABASE tableName --" or "; DROP TABLE tableName --." This will mess up the system, as it will delete the database or the table respectively. This is not traceable to a user in particular. However, it might be traceable to an IP address. The attacker might use a public library computer which cannot be tracked to a specific user, or some sort of other public computer on a public network.

In this case, the user was not logged in, and therefore no accounts can be tagged to them. Only the IP address of a public computer is logged. However the hacker can easily deny their actions.

A good way to avoid SQL injections is using libraries for PHP like PDO. By using the statement prepare, you are most likely avoiding injections(Marcus, 2017). Another approach is checking every single form field for string characters like ";" or its ASCII alternative. It is very important to always check the input for code. If manually checking for code, create a small allow list of characters that can be used, such as a-z, A-Z, 0-9, and non-harmful non-alphanumerical characters.

The site also sells tapir merchandise. As the system might use HTTP, Mal can intercept packages of purchases and store them for a while. Then he can send these packets to Tapirs

Unlimited dozens of times. Money will be debited from victims' credit cards, purchases will be made, and Mal cannot be identified. He is just sending along packets he received (and stored) a while ago. This will damage the trust of users on the Tapirs Unlimited website. Mal is just a computer on the network, it is unlikely that the changes are tagged to his machine. Especially if the source address in the IP packet is from an innocent user's machine.
A way to prevent this type of attack is to use HTTPS exclusively, and at each new connection use a different key. This way, Mal won't know the content of each packet. Even if he did, when doing the replay attack the key will no longer be in use.
This is a repudiation attack as Mal is attacking the system, and Mal's actions can't be linked to him.

I - Information disclosure

The site uses HTTP, which is not encrypted. So files and text will be sent over the network via plain text, which is a huge security risk. If users access TU via HTTP, all the information they send to and receive from the website will be visible to eavesdroppers. Whether the information is username and password, credit card information, or something else it should not be visible to eavesdroppers.
As eavesdroppers are getting access to classified information this is a case of information disclosure attack.
Mitigation: Do not allow traffic from HTTP, only allow traffic from HTTPS. This way all the communications are encrypted. So, it will be way harder to eavesdrop. It is also important, if doing the Diffie-Hellman key exchange, to demand that the numbers used for the key are appropriate. What's more, b should be randomly generated at each communication, so the key changes at each interaction.

D - Denial of service

The most obvious Denial of Service attack is DDoS (Distributed Denial of service). A bunch of thermostats, smart fridges, and other devices (zombies) access Tapirs Unlimited at the same time. This way they can overwhelm the network. So, the network will have a lot of traffic, and the service may get considerably slower. In the worst case, real users might not be able to access TU at all (which is the main goal of the attack) (According to this article).
In other words, in a DDoS attack, numerous zombies try to access a website at the same time, causing considerable network traffic, so that real users are not able to access the website. So for some minutes, users do not have access to the website.
Mitigation: Block high amounts of traffic from a single IP address or IP range. And check for users who have a single behavioral profile, such as device type, geolocation, and browser version. Look for odd traffic patterns, like a traffic spike every 7 minutes. Block said unusual traffic. This will hopefully prevent the attack or at least mitigate the damage.
There are also some bots and tools that can be used to protect the server against DDoS such as this one and this one.
Even if it is not possible to block DDoS completely, try to mitigate it. Make it so that the service is only unavailable for a couple of seconds.

E - Elevation of privilege

Arthur Viegas Eguia

There are a few attacks that come to mind here. One is based on the fact that the website uses HTTP as well as HTTPS. In this first hypothetical attack, just let an eavesdropper listen to the communications between the users and the server. As soon as someone does administrator tasks (changes or deletes something posted by somebody else, bans users, sees confidential information, …), the eavesdropper can get the username and password of the administrator from their previous packets. If the eavesdropper has this information in hand, they can hack into the administrator account and do whatever they want, like delete tables, ban users, arbitrarily delete posts, and steal information.

Another way to hack into the administrator's account is a bit harder. Start out by doing a series of injection attacks and discovering what information is returned from the queries. Just like the system might be vulnerable to "; DROP TABLE --" it might also be vulnerable to "; SELECT --." Find the name of all tables, find the tables that are likely to store administrator accounts, and add rows to those tables with the information of a new account. With this account in the database, the attacker can access the system with administrator privileges.

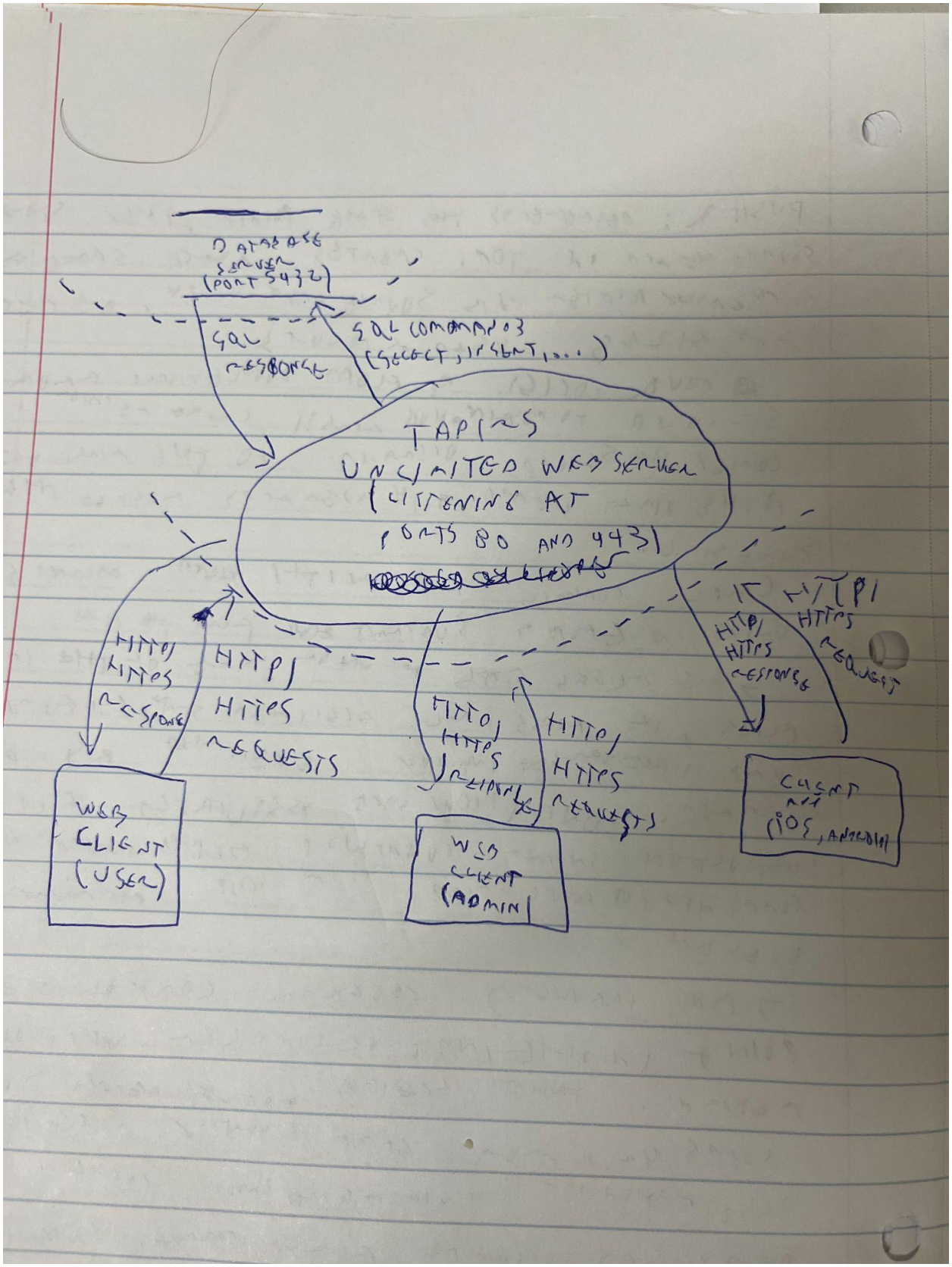
A third way to gain administrator privilege is through a modified version of the replay attack. Whenever an administrator is doing actions over HTTP that are restricted to administrators (banning people, getting information,...), the attacker can save the packages sent by the administrator and do the replay attack, one (or a few) times. Even worse, as the packet is in HTTP the attacker can modify the plain text in it. This way, the attackers can send the modified packets to TU so the system bans different users, or deletes different tables.

A way to prevent the attacks described above is to use HTTPS exclusively, using a secure encryption algorithm and key. What's more, the Symmetric Encryption Key should be different every time, so the replay attack can be prevented.

The other mitigation strategy is checking for code in forms and treating all input data to prevent SQL injections (or any other type of code injections). According to Marcus, libraries such as PDO do this pretty well. But a similar system can be coded by the Tapirs Unlimited team.

By using HTTPS and preventing injections these issues can be mitigated.

Arthur Viegas Eguia



Arthur Viegas Eguia

Sources:

Bot mitigation. CHEQ Essentials -. (n.d.). https://essentials.cheq.ai/bot-mitigation-acq/?r=gads&utm_source=google&utm_medium=cpc&utm_campaign=Search_TCPA_USA_PM%2BBM_Desktop&utm_content=656918743662&utm_term=bot+mitigation+solution&adgroupid=149707238818&matchtype=b&extensionid=&cq_src=google_ads&cq_cmp=20029007264&cq_con=149707238818&cq_term=bot+mitigation+solution&cq_med=&cq_plac=&cq_net=g&cq_plt=gp&hsa_acc=8098481974&hsa_cam=20029007264&hsa_grp=149707238818&hsa_ad=656918743662&hsa_src=g&hsa_tgt=kwd-1021601444871&hsa_kw=bot+mitigation+solution&hsa_mt=b&hsa_net=adwords&hsa_ver=3&gclid=CjwKCAjwvrOpBhBdEiwAR58-3MK_T7HsoJy-FW2BFwDNuPpooC59lpOjyRJ8KRvwuw7J4TBiEhZvsBoCOKEQAvD_BwE

DDoS attack protection and mitigation | Akamai. (n.d.-a). https://www.akamai.com/products/prolexic-solutions

Marcus, D. (2017, November 26). *PHP PDO prepared statements tutorial to prevent SQL injection*. Website Beaver. https://websitebeaver.com/php-pdo-prepared-statements-to-prevent-sql-injection

What is a distributed denial-of-service (ddos) attack? - cloudflare. (n.d.-b). https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/