

Forum:[Hack Forums](#)

Thread: [Mega SQLi & MITM Defacement Guide](#)

Tyr - February 19, 2020, 8:01 pm

Spoiler ([Click to View](#)) 2/19/2020 - 10:46 PM - Thread Under Construction; beta layout & information. Please excuse any errors made within the guide. This thread is also strictly for educational purposes and I am not responsible for any actions on your behalf.

Introduction

Hello everyone. After browsing for quite a few hours, I noticed there isn't a mega-guide for web penetration, exploitation, and ultimately yes, site defacement. After some tedious browsing I saw that there was a thread created years ago explaining the process of these techniques briefly. With this being said, I would like to share some of the knowledge I have gained over the years being a Sysadmin for private gaming network(s), as well as the Whitehat knowledge I have accumulated to create a thread fitting for 2020, and bring to light more to this thread than just "defacing a website."

I see there was a popular thread here posted quite some time ago by a closed account that had some great information in it, but it was very outdated and not detailed enough. Not only that, but I see a lot of newer members asking questions and not getting the information they requested or simply posting for the sake of "needing a website hacked."

As for a little background for me, I have been involved with designing queries involving SQL and sport fairly comprehensive knowledge of database architecture in general. I also have accumulated a fair amount of experience with C#, C++, Python, Java, and PHP for around 7-8 years on and off. At the age of 13, I became interested in making money online and was a member of a SEO board and coding for multiple projects involving MMO framework for many years prior to migrating here.

New Age of Website Hacking

With time, people, software designers, and companies learn. In the old days, you used to be able to find vulns within sites quite easily, especially when they were not secured. Cybersecurity has now advanced to the point where things are much more complex.

A study was made that stated that there is an attack every 39 seconds on average on the web and the non-secure usernames and passwords that are being used give attackers more chance of success.

(Source: Security Magazine)

An attack on a website does not always mean it has been hacked. For example, thousands of attacks are logged within larger websites and monitored and filtered out by a firewall system. These attacks are logged and monitored by said systems and the web application firewall on the website is in place to make sure the attacks won't be successful. The majority of these attacks are in reference to DDoSing. I will not go far into explaining the specifics of how a DDoS attack can help you due to forum policies though.

A common new age attack, especially with companies with poor security is known as a MITM (Man in the Middle) attack. Let me explain how this works. In a man-in-the-middle attack, attackers places themselves between two devices (often a web browser and a web server) and intercept or modify

communications between the two. The attackers can then collect information as well as impersonate either of the two agents. In addition to websites, these attacks can target email communications, DNS lookups, and public WiFi networks. If you are smart and/or have knowledge about the potential impact and relation to website defacement and hacking, then you will be able to see the comparison I am making here.

So how can this help me deface a website? How does this play out? Let's say you received an email that appeared to be from your VPS, asking you to log in to your account to confirm your email for "the new year of 2020." The victim clicks on a link in their email, skype, Discord, etc. And they then are taken to what appears to be your VPS's website, where you log in with your credentials and perform the requested task. When in fact, this is simply a replica of the site.

In such a scenario, the man in the middle (MITM) sent you the email or link making it appear to be legitimate. He also created a website that looks just like your VPS's website, so you wouldn't hesitate to enter your login credentials after clicking the link in the email. But when you do that, you're not logging into your VPS, you are basically giving the account VPS info over to you, the man in the middle. From there, you have full reign over their account and can then gain access to SSH and pretty much hi-jack their website.

Most hackers will then use this to their advantage to subject the VPS owner or even personal computer user to pay them in order to regain access to the account. Due to HF's rules in regards to Blackhat activity, I am not quite sure if I can go into much more detail other than provide you with a general idea on how this works above. Phishing is fraud and this portion of the guide is strictly for educational purposes only to make you aware. Many Anti-viruses have caught on to this over the last decade and have deployed countermeasures to stop these attacks, therefore unless you know what you are doing and aren't willing to risk possible legal ramifications you should skip ahead to the rest of the guide where I will cover older, yet effective techniques.

SQL Injection

Now onto the fun stuff! SQL injection is quite simple to understand and is newbie friendly. It is an SQL code injection technique, used to attack a web database by inserting malicious SQL statements. SQL injections are typically performed on faulty and poorly designed CMS's. Typically, the site will have a lot of extensions and such that are not properly secured or simply poorly developed. SQLi is a simple but powerful technique that can compromise a website's entire database. Here is a diagram that basically explains just how powerful and underestimated SQL Injection is in 2020.

[UNDER CONSTRUCTION; UPLOAD ISSUES]

As you can see from the above diagram, there are many ways one can perform an SQLi attack. For the sake of keeping this noob friendly, I will provide an example of what is typical for an SQLi attack. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL queries and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is then executed.

Let's suppose that we have a web application that takes the parameter article via a \$_GET request and queries the SQL database to get article content. By evaluating the article parameter, you can see it is assigned to \$article variable without any sanitization or validation. Go ahead and analyze the parameters and variables. For example:

UNDER CONSTRUCTION; UPLOAD ISSUES

Bingo. There is an article with an ID of 1, and 1 equals to 1 which is true. I think right? I don't know, it's been awhile since I took 2nd grade math. Because of this knowledge, you can now adjust the string yourself. There can be various other username and password combinations to play with the vulnerable sites. You can even create your own new combinations for the site login page. I will get to finding that information later in the guide.

Some sample combinations can be:

UNDER CONSTRUCTION; UPLOAD ISSUES

Entry Manipulation

Let's say you have identified a plugin endpoint that accepts the user ID via a `$_GET` request and displays their user name. The first thing we will do is to manipulate the entry point (user input: `$_GET` parameter) and observe the response. There are many ways you can detect SQL vulns. Typically what we are looking for is to see if our input causes the output to change in any way at all. This can include missing headers, broken images, MySQL errors, blank pages, and much more. Ideally, we want to see an SQL syntax error which could indicate that our input is parsed as part of a query. If a MySQL syntax error pops up, you have struck gold.

Another very common way of checking for a vulnerability is to simply add a ' at the end of the URL. This is one of the most rudimentary techniques for checking for a vulnerability. Keep in mind that this method is fairly outdated for finding vulns.

Parameterized Data & Its Importance

Parameterized data as such can be manipulated by a skilled and determined web exploiter with knowledge of SQL. The injection process works by prematurely terminating a text string and appending a new command. Because the inserted command may have additional strings appended to it before it is executed, the malefactor terminates the injected string with a comment mark `--`. Subsequent text is ignored at execution time.

The injection process works by prematurely terminating a text string and appending a new command. Because the inserted command may have additional strings appended to it before it is executed, the malefactor terminates the injected string with a comment mark `--`. Subsequent text is ignored at execution time.

Parameterized data in Load Impact is implemented using something called data stores. A data store is basically a table with data that can be shared by multiple clients in a load test. The usage is simple: You create a new data store in the user scenario configuration interface, assign a name to it, then upload a text file with the data you want to insert into it.

By constant trial and error, using the above methods combined, and some useful programs you will find at the end of the thread, you will be able to do this all quite easily and bypass or simply even get the user name and password for the MySQL server. For legal reasons, I am going to skip this part of obtaining the username, but by utilizing the above information, you should be able to gain access.

Cracking the Hashed Password

This can be one of the trickiest parts of website hacking. Hashing is a one-way process. An experienced hacker can crack a password-list by regenerating the hashes and comparing them to the stored hash you have found within MySQL to crack the actual password.

This site <https://crackstation.net/> attempts to do this for you - it runs through passwords lists and tell you the cleartext password based on your hash. I highly recommend it for anyone looking to decrypt

hashes within MySQL. There are also many other sites.

[b]Compiling Your Defacement[/b]

Once you have gained access and would like to upload your defacement, first you must create it. This can be done easily with basic HTML or PHP knowledge. The way you code your defacement is entirely up to you and there are many resources here on HF where you can find pre-coded defacement resources. However, I recommend you create your own. The possibilities are endless and you can quite literally lock the entire website. Again, I will not cover too much of this because it is fairly straightforward and it is also something quite Blackhat in which I do not want to affiliate myself with.

Uploading Your Defacement

Now that you have generated your code, you have it saved, and edited make sure to change it to an .html file, or else it won't work on certain browsers. Open up your favorite FTP service and connect to the FTP host which can usually be found somewhere on the website's control panel. Upload the file to public_html and make sure you delete any files that can be used to trace the origin of your attack. You will then have to CHMOD the permissions. After this go to the page in which you uploaded your defacement and then you should see your uploaded code in the form of an HTML defacement.

Ending Remarks, Useful Tools, etc.

Please keep in mind that defacing websites, especially for monetary gain is extremely illegal and especially when it comes to doing it via MITM (the most savage method in my opinion). I would like to thank the users who have helped contribute to this thread by providing quite a lot of data in which I have tried my best to scale up to 2020 standards. This thread was created purely for educational purposes only and testing purposes.

Here are some excellent tools that will make all of this much easier for you:

Spoiler [\(Click to View\)http://sqlmap.org/](http://sqlmap.org/)
<https://suip.biz/?act=sqlmap>
<https://www.softpedia.com/get/Internet/Servers/Database-Utils/Data-Xtractor.shtml>

Here is a thread via HF with some incredibly useful information in regards to bypassing logins:
<https://hackforums.net/showthread.php?tid=45621>

HF User credits, thank you for your help and contributions: nightkid411; Doga; Jaystar; PinningYou; and everyone who posted information within this section.

Sincerely,
Fad3d

Money- - February 21, 2020, 12:42 am

Good stuff. I rarely see some genuine threads over here. Mostly just database injection requests and copy-paste tutorials from other forums or blogs.

If you can include several techniques such as Injection on the User-Agent Header or Cookie Header.

As well as several database injections like Oracle, MongoDB and several other not common DBs instead of MSSQL and MYSQL will be good and stand out.

Don't forget to explain or show an example of a vulnerable SQL back-end statement.

To those getting into SQLI, highly recommend you to understand SQL statements and having some web development experience.

It will definitely help you gaining an insight and full understand of the website infrastructure and layout.

Instead of just blindly copying SQLI payloads into the URL parameters or input fields.

Tyr - February 21, 2020, 1:30 am

(Feb 21, 2020 01:42 AM)Money- Wrote: [\[CITE\]](#) Good stuff. I rarely see some genuine threads over here. Mostly just database injection requests and copy-paste tutorials from other forums or blogs. If you can include several techniques such as Injection on the User-Agent Header or Cookie Header.

As well as several database Injections like Oracle, MongoDB and several other not common DBs instead of MSSQL and MYSQL will be good and stand out.

Don't forget to explain or show an example of a vulnerable SQL back-end statement.

To those getting into SQLI, highly recommend you to understand SQL statements and having some web development experience.

It will definitely help you gaining an insight and full understand of the website infrastructure and layout.

Instead of just blindly copying SQLI payloads into the URL parameters or input fields.

Hey there, thank you for the compliment! And yes, this entire portion of the forum made me a little upset, because the majority of posts are from brand new accounts asking the silliest questions ever. I hope that people new to the website hacking game will read the entire thing properly, because there's useful information that is important to not miss out on. I will definitely be adding an entire section for Cookies and I will provide some information in regards to Oracle, but I do not know much about the structure of MongoDB actually so if you have anything you'd like to see added, definitely let me know.

Also, yes, understanding statements is really where one should start. I am going to actually add this here and eventually to the thread once I wake up, but you reminded me of a great resource for those new to SQL and how exactly it functions in correspondence to the webserver and overall website in general. There's practice tests, a practice area for your own personal use for writing queries, and much more involving other languages.

<https://www.w3schools.com/sql/default.asp>

<https://www.w3schools.com/css/default.asp>

<https://www.w3schools.com/php/default.asp>

All three of these links will teach newer coders (even intermediate ones such as myself because of the practice areas) very important fundamentals about the stuff discussed within the guide. This is probably one of my favorite sites for learning to code. I'd recommend everyone check it out.

Quote:To those getting into SQLI, highly recommend you to understand SQL statements and having some web development experience.

It will definitely help you gaining an insight and full understand of the website infrastructure and layout.

Instead of just blindly copying SQLI payloads into the URL parameters or input fields.

Definitely something that I truly do recommend anyone reading this guide to do. Back in the day you used to be able to do a blind injection quite easily but it's hardly ever effective on a lot of larger websites unless they severely messed up or added nulled or outdated plugins and such.

ALSO: One last thing. What do you think about the potential MITM attack section of this thread? It's a little different, which is why I wanted to add it, but it's also highly advanced and I do not think most members here on this forum will be able to grasp just how powerful you can be by being the "man in the middle."

I recently found out about it a year ago or so when I had it happen to me actually. Most use it for ransom purposes but it's also a tactic that cyber security experts use in order to trap major blackhat hackers by falsifying their key. It's so simple, yet so fucking brilliant.

Perica - February 22, 2020, 4:24 am

Please more tutorials like this. Also you can write common attacks on website.

Tyr - March 1, 2020, 11:36 pm

(Feb 22, 2020 05:24 AM)Perica Wrote: [\[CITE\]](#) Please more tutorials like this. Also you can write common attacks on website.

I'll do my best, website and forum hacking is what got me into delving deeper into my computer when I was young and I am still adjusting to the advancements in cybersecurity and AV's after taking a long break.

I'll be releasing a guide exclusive to MITM within the next month with far more detail and alongside it offering very cheap paid mentorship for anyone interested in learning more about MITM attacks and how to carry one out; as it typically isn't an exact science and there are many techniques to executing the attack involving a few different areas of infiltration in general.

Keep in mind it will be more for the "mobile laptop hacker" if you catch my drift, but can still be accomplished halfway across the world, that's the beauty of it.

Of course everything is for educational purposes only. I'll come back with a link when it is complete.