



# SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

Web Security - (IE2062)

Assessment – Web Audit

Domain – [www.tesla.com](http://www.tesla.com)

Name	Wimalasena G.R.T.D
Index Number	IT20059354
Date of submission	4 <sup>th</sup> Nov 2021

# Acknowledgement

It is not important to memorize some theories in order to learn. A good learner should always put what they've learned into practice. When it comes to a profession like cybersecurity, this is a requirement.

We learned a number of ideas in the Web Security module. However, it's crucial to see how those theories apply to a real-world online application, and this project, which is based on a web audit, helped us understand the practical side of web application penetration testing.

I'd want to give my deepest appreciation to the module's instructor, Dr. Lakmal Rupasinghe, Ms. Chethana Liyanapathirana, and other assistant lecturers who led and advised me throughout the process.

# Abstract

Cybercrime, deception, and data breach are all risks that pose major risks to businesses. A great deal has been lost, and organizations must devise procedures to prevent the risks from becoming serious and to prevent similar disasters. This investigation looked into the mechanisms connected with IT security web audits and how they may help firms enhance their IT security. The study assessed IT administrators' and employees' awareness of cybercrime risks, as well as their understanding of IT security audit norms and standards and the impact of IT security audit on the organization's growth.

This inquiry is using an organization as a context, analyzing the organization's flow IT security audit level and evaluating adaptability for the establishment of an IT security audit strategy and system. A quantitative investigation was carried out in order to gather more information on cybercrime and to compile more comprehensive data on the subject. This inquiry definitely demonstrated that an IT security audit is critical for the growth of any activity that supports technology.

# Introduction

Cybercrime is a threat that every business face, and there is growing concern about the most effective way to combat it. Every day, cybercrime causes harm on organizations' data systems, resulting in physical pain, financial loss, and reputational risk. There's no room for vanity here. Then, in accordance with the business rules and standards, an IT web audit program for those frameworks is implemented, and an audit report is issued that includes the findings and the next steps that should be taken to mitigate any data security risks. In fact, IT auditing requires significant resources like time and money. In any event, the cost of a cybercrime, deception, or data breach is expected to be significant.

As a result, it pays to avoid it. The main purpose of this analysis is to present the relevance of online security auditing and to examine the benefits of IT security auditing as a useful tool for improving an organization's data security. The research also looks at the organization's attention to cybercrime risks, how well they use international security norms and rules, and how they conduct periodic IT security audits.

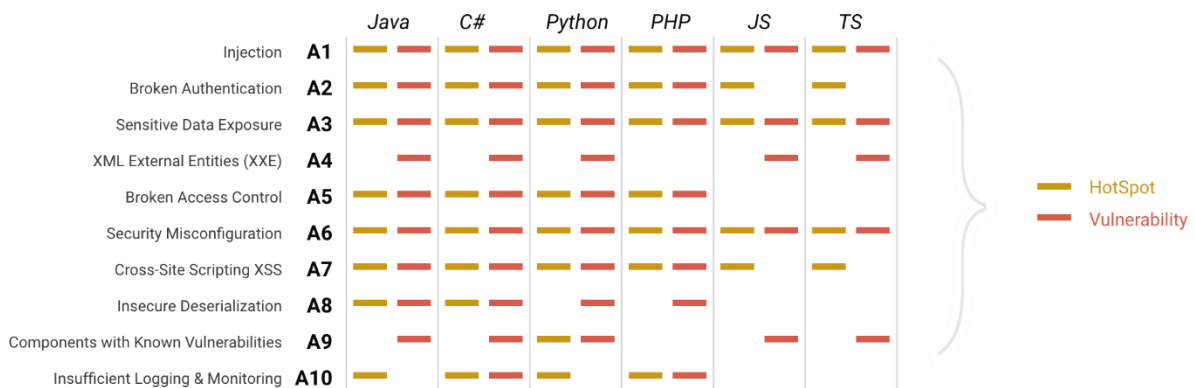
## Objective of the Audit

The vulnerability assessment for <https://www.tesla.com/> is performed in collaboration with the Web Security module's second year second semester assignment. The goal of the audit is to identify as many vulnerabilities as possible within the target scope, classify them according to risk categories, and report them.

## Introduction to OWASP top 10

The Open Web Application Security Project is a non-profit organization that maintains a list of the most common security vulnerabilities found in websites in order to assist developers and security researchers in creating more secure websites that are less vulnerable to cyber assaults. According to <https://owasp.org/www-project-top-ten/>, the following are the OWASP top ten:

- I. Injection
- II. Broken Authentication
- III. Sensitive data exposure
- IV. XML External Entities (XXE)
- V. Broken Access control
- VI. Security Misconfiguration
- VII. Cross-Site Scripting
- VIII. Insecure Deserialization
- IX. Components with Unknown Vulnerabilities
- X. Insufficient Logging and Monitoring



## **Injection**

Injection problems arise when a user's input is interpreted by the program as genuine instructions or credentials. These attacks are very common, and they are very dependent on the nature technology and how they interpret user input. The following are some of the most common injection fits:

- SQL Injection
- Command Injection

## **Broken Authentication**

In today's online applications, user authentication is essential. Authentication systems verify the identity of the users. An attacker who discovers a security hole in an authentication method can spoof a legal user of the system. The following are some examples of common authenticating attacks:

- Busing weak credentials
- Weak session cookies
- Username and Password Brute forcing

## **Sensitive Data Exposure**

Web apps may unintentionally expose sensitive information about their users. This might include data like names, phone numbers, and emails that aren't significant. However, very confidential material such as usernames and passwords can be publicly released. These data may be stolen, deleted, or modified by the attacker.

## **XML External Entities (XXE)**

An XXE attack exploits features of XML parses and data. An attacker will be allowed to interact with the backend or external systems that the program may access, as well as read files on such systems. XXE assaults can also lead to denial-of-service attacks, server-side request forgery attacks, port scanning, and remote code execution.

## **Broken Access Control**

There are certain pages that are password-protected and accessible only to frequent visitors. If a website visitor can view protected pages that they are not permitted to see, the access restrictions have been breached.

- The disclosure of sensitive information as a result of this type of attack.
- Gain access to features that aren't supposed to be there.

## **Security Misconfiguration**

When security setups are not adequately addressed, misconfigurations arise. The following are some examples of common security misconfigurations:

- Default accounts, which have the same usernames and passwords as before.
- HTTP security headers are not utilized, or the server HTTP header contains too much information.
- Extensive error messages that enable an attacker to learn more about the target system.
- Unnecessary features such as services, pages, accounts, and privileges are enabled.
- Permissions for cloud services, such as S3 buckets, are inadequately set.

## **Cross-Site Scripting**

Because of single user inputs, XSS is a sort of injection attack that allows an attacker to run malicious scripts on a victim's machine. XSS attacks are more likely to be launched through web development languages like Javascript, VBScript, Flash, and CSS. There are three types of XSS attacks:

- Stored XSS
- Reflected XSS
- DOM-Based XSS

## **Insecure Deserialization**

When an application's logic is exploited by accessing untrusted data, insecure decoding occurs. In other words, malicious code replaces the data handled by an application, allowing the attacker to execute anything from denial of service to remote code execution in order to gain access to the system. Any application that saves or retrieve information and lacking validations and integrity checks, such as e-commerce sites, forums, APIs, and application runtimes (Tomcat, Jenkins), is vulnerable to these kinds of attacks.

## **Components with Known Vulnerabilities**

If the website's underlying applications are outdated, there's a good probability that a well-known vulnerability will be discovered, which may be exploited to get access to the system. As a result, as an attacker, you just have to do extremely basic work, which is why OWASP has given this vulnerability a low rating.

## **Insufficient Logging and Monitoring**

When web applications are built, user actions should be tracked. This is important because, in the case of an incident, the attacker's actions may be traced. If an attacker gains access to a system and there is no logging mechanism in place, it is impossible to identify the impact of the attacker's actions. The biggest consequences of this vulnerability are regulatory ramifications and the danger of subsequent assaults. To solve this vulnerability, intrusion detection and prevention systems can be used, and logs should include the following information:

- HTTPS status codes such as 200 and 403
- Usernames
- IP addresses
- Time stamps on tracked actions

## About the target

Tesla, Inc., located in Palo Alto, California, is an American electric vehicle, sustainable energy, and technology company. Tesla creates electric vehicles, battery energy storage from the house to the grid, solar panels and solar roof tiles, as well as other goods and services. Tesla sold the most battery electric and plug-in electric vehicles in 2020, accounting for 16 percent of the plug-in market (which includes plug-in hybrids) and 23 percent of the battery-electric (purely electric) market. The firm produces and installs solar systems in the United States through its subsidiary Tesla Energy.

## Assessment Scope

Scope of the security audit according to <https://www.tesla.com/> is,

- \*.teslamotors.com

- \*.tesla.services

- \*.teslainsuranceservices.com

- \*.solarcity.com

Any host verified to be owned by Tesla Motors Inc. (domains/IP space/etc.)

Official Tesla Android apps

Official Tesla iOS apps

## Out of Scope

- feedback.tesla.com

- feedback.teslamotors.com

- ir.teslamotors.com

- mkto.teslamotors.com

- shop.eu.teslamotors.com

Any other third-party websites hosted by non-Tesla entities



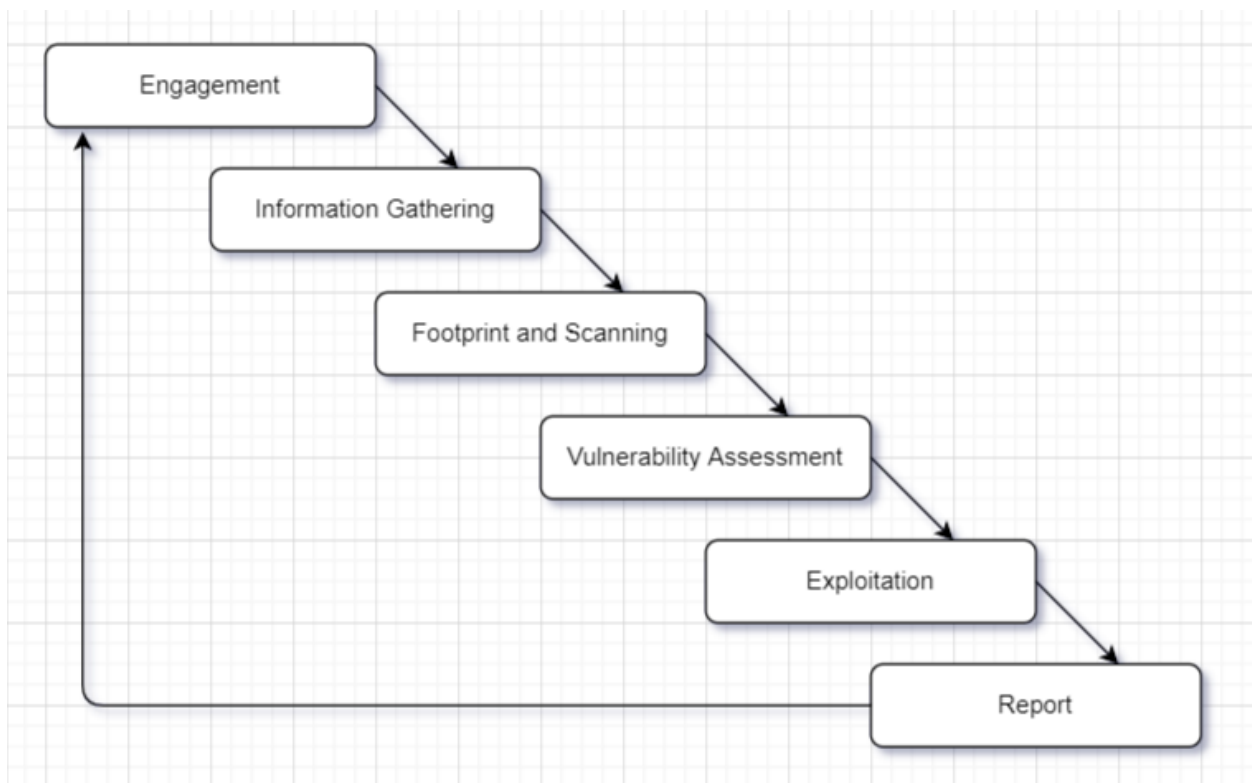
employeefeedback.tesla.com

## Focus Area

- Tesla's public facing web application
- Vulnerabilities in other applications owned by tesla
- Vehicle or product related issues

## Assessment Methodology

In order to get the most out of a vulnerability assessment or penetration test, there is a certain set of processes that must be performed. Otherwise, the tester's results will be screwed up, and important vulnerabilities would be missed. This analysis is performed in accordance with a professional penetration test's standard procedure.



## Information Gathering

The penetration tester gets all publicly accessible information about the online application from a variety of sources at this stage. It helps in the penetration tester's understanding of the online application and network environment. This recently bought information helps the tester in identifying vulnerabilities and determining their effect.

There are two methods of data collection. They are as follows:

1. Passive - gathering data without interaction with the target web application
2. Active - collecting data while interacting with the target web application

Because this is a contactless information collecting technique between myself and the target, the major focus of this project is on passive information gathering techniques. There are a few processes that should be followed when acquiring passive data. Target validation, subdomain discovery, vulnerability screening, and fingerprinting are some of them.

## Finding Subdomains

Subdomain identifier is an important part of a website vulnerability assessment because there will be additional websites, login forms, test sites, and developer sites associated with our main target, some of which might contain vulnerabilities that can be exploited to gain access to our target system. **Model3.tesla.com** is an example of a subdomain that is unfinished and has experimental features. As a result, there's a good probability they'll include exploits.

Popular tools that are developed to find subdomains are:

- Sublist3r
- Crt.sh
- Nmaps

And also there are many of web sites which gives all of the subdomains we can access by searching. For this audit I have used Sublist3r and crt.sh.

## Sublist3r

Sublist3r is a python-based program for detecting a website's subdomains. In order to execute a penetration test or an attack, attackers and penetration testers use this program to collect subdomains of the targeted domain.

So, you need to git clone the tool first. It can be cloned from the official git repository, <https://github.com/about3la/Sublist3r>

For installing it into kali, we must run “**git clone https://github.com/about3la/Sublist3r**” in the terminal. Then we must go inside the directory and compile the sublist3r.py with mentioning the targeting domain at the end.

```
python3 Sublist3r.py -d tesla.com
```

```
(tharindu@kali) - [~/Desktop]
$ cd Sublister

(tharindu@kali) - [~/Desktop/Sublister]
$ ls
LICENSE  MANIFEST.in  README.md  requirements.txt  setup.py  subbrute  sublist3r.py

(tharindu@kali) - [~/Desktop/Sublister]
$ python3 sublist3r.py -d tesla.com

          SUBLIST3R
          # Coded By Ahmed Aboul-Ela - @about3la

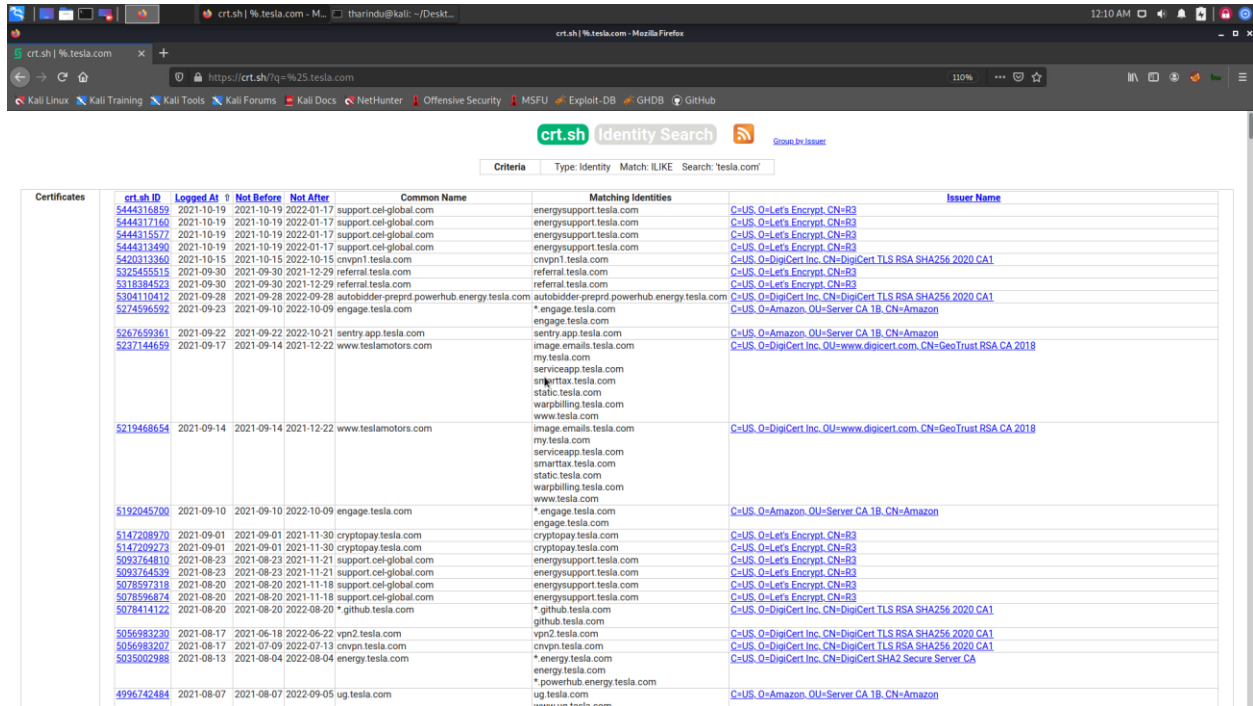
[-] Enumerating subdomains now for tesla.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
```

Sublist3r could find 324 unique subdomains for given domain “tesla.com” within few seconds. How incredible!

```
[ - ] Total Unique Subdomains Found: 324
www.tesla.com
3.tesla.com
akamai-apigateway-automation.tesla.com
akamai-apigateway-bender.tesla.com
akamai-apigateway-captiveunderwriting.tesla.com
akamai-apigateway-crawford-prd.tesla.com
akamai-apigateway-crawford-stg.tesla.com
akamai-apigateway-deliveryopsapi.tesla.com
akamai-apigateway-deliveryopsapi1.tesla.com
akamai-apigateway-dev-warptmsapiserver.tesla.com
akamai-apigateway-einvoicing.tesla.com
akamai-apigateway-finplat-prd.tesla.com
akamai-apigateway-finplateng.tesla.com
akamai-apigateway-finplateng-routeone.tesla.com
akamai-apigateway-fta.tesla.com
akamai-apigateway-inventorytxnextapi.tesla.com
akamai-apigateway-logisticsratesapi.tesla.com
akamai-apigateway-materials.tesla.com
akamai-apigateway-mfs-supplier.tesla.com
akamai-apigateway-ops-warp3pl.tesla.com
akamai-apigateway-payment.tesla.com
akamai-apigateway-prd-global-deliveryopsapi.tesla.com
akamai-apigateway-procuretopayapi.tesla.com
akamai-apigateway-profileapi.tesla.com
akamai-apigateway-qa-captiveunderwriting.tesla.com
akamai-apigateway-scraprecycle.tesla.com
akamai-apigateway-shipmentplanningapi.tesla.com
akamai-apigateway-stg-bender.tesla.com
akamai-apigateway-stg-captiveunderwriting.tesla.com
akamai-apigateway-stg-deliveryopsapi.tesla.com
akamai-apigateway-stg-deliveryopsapi1.tesla.com
akamai-apigateway-stg-finplateng-defi.tesla.com
akamai-apigateway-stg-finplateng-routeone.tesla.com
akamai-apigateway-stg-fta.tesla.com
akamai-apigateway-stg-ops-warp3pl.tesla.com
akamai-apigateway-stg-packaging2.tesla.com
akamai-apigateway-stg-procuretopayapi.tesla.com
```

## Crt.sh

It is just a website which make works easier, and it gives deeper information rather than Sublis3er. We can get even 4<sup>th</sup> level subdomains through this web site.



The screenshot shows the crt.sh Identity Search interface. The search criteria are set to 'Identity' and 'Match: ILIKE' with the search term 'tesla.com'. The results are displayed in a table with columns for Certificate ID, Logged At, Not Before, Not After, Common Name, Matching Identities, and Issuer Name. The table lists various certificates issued to tesla.com and its subdomains, including support.cel-global.com, energysupport.tesla.com, and others. The certificates are issued by different issuers, including C=US, O=Let's Encrypt, and C=US, O=DigiCert Inc.

Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
	54443116559	2021-10-19	2021-10-19	2022-01-17	support.cel-global.com	energysupport.tesla.com	C=US, O=Let's Encrypt, CN=B3
	54443121160	2021-10-19	2021-10-19	2022-01-17	support.cel-global.com	energysupport.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5444315577	2021-10-19	2021-10-19	2022-01-17	support.cel-global.com	energysupport.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5444313490	2021-10-19	2021-10-19	2022-01-17	support.cel-global.com	energysupport.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5420312360	2021-10-15	2021-10-15	2022-10-15	cnvnp1.tesla.com	energysupport.tesla.com	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
	5325455515	2021-09-30	2021-09-30	2021-12-29	referral.tesla.com	referral.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5318384523	2021-09-30	2021-09-30	2021-12-29	referral.tesla.com	referral.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5304110412	2021-09-28	2021-09-28	2022-09-28	autobidder-preprd.powerhub.energy.tesla.com	autobidder-preprd.powerhub.energy.tesla.com	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
	5274596592	2021-09-23	2021-09-10	2022-10-09	engage.tesla.com	*engage.tesla.com engage.tesla.com	C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
	5267659361	2021-09-22	2021-09-22	2022-10-21	sentry.app.tesla.com	sentry.app.tesla.com	C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
	5237144659	2021-09-17	2021-09-14	2021-12-22	www.teslamotors.com	image.emails.tesla.com my.tesla.com serviceapp.tesla.com myrttax.tesla.com static.tesla.com warpbilling.tesla.com www.tesla.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=GeoTrust RSA CA 2018
	5219468654	2021-09-14	2021-09-14	2021-12-22	www.teslamotors.com	image.emails.tesla.com my.tesla.com serviceapp.tesla.com myrttax.tesla.com static.tesla.com warpbilling.tesla.com www.tesla.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=GeoTrust RSA CA 2018
	5192045700	2021-09-10	2021-09-10	2022-10-09	engage.tesla.com	*engage.tesla.com engage.tesla.com	C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
	5147208970	2021-09-01	2021-09-01	2021-11-30	cryptopay.tesla.com	cryptopay.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5147209272	2021-09-01	2021-09-01	2021-11-30	cryptopay.tesla.com	cryptopay.tesla.com	C=US, O=Let's Encrypt, CN=B3
	50931764810	2021-08-23	2021-08-23	2021-11-21	support.cel-global.com	energysupport.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5093764532	2021-08-23	2021-08-23	2021-11-21	support.cel-global.com	energysupport.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5078597318	2021-08-20	2021-08-20	2021-11-18	support.cel-global.com	energysupport.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5078596874	2021-08-20	2021-08-20	2021-11-18	support.cel-global.com	energysupport.tesla.com	C=US, O=Let's Encrypt, CN=B3
	5078414122	2021-08-20	2021-08-20	2022-08-20	*github.tesla.com	*github.tesla.com github.tesla.com	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
	5056983230	2021-08-17	2021-06-18	2022-06-22	vpn2.tesla.com	vpn2.tesla.com	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
	5056983207	2021-08-17	2021-07-09	2022-07-13	cnvnp1.tesla.com	cnvnp1.tesla.com	C=US, O=DigiCert Inc, CN=DigiCert TLS RSA SHA256 2020 CA1
	5035002988	2021-08-13	2021-08-04	2022-08-04	energy.tesla.com	*energy.tesla.com energy.tesla.com	C=US, O=DigiCert Inc, CN=DigiCert SHA2 Secure Server CA
	4996742484	2021-08-07	2021-08-07	2022-09-05	ug.tesla.com	*powerhub.energy.tesla.com ug.tesla.com www.run.tesla.com	C=US, O=Amazon, OU=Server CA 1B, CN=Amazon

Do not forget to save all those subdomains to a text file for later use purposes.

## Finding alive Subdomains

As I mentioned before some of these subdomains might be no alive or not in use. They might create by developers for some testing purposes. So, we must use a tool to filter out the alive domains rather than doing it manually to get a result within a few seconds. We need to install httprobe tool which we can find it on GitHub. If it is not pre-installed. <https://github.com/tomnomnom/httprobe>

In this step we are using previously saved subdomains.txt file to get the alive subdomains.

```
(tharindu@kali) - [~/Desktop]
$ cat subdomains.txt | httprobe >> alive.txt
```

The results of httprobe are stored in a text file called alive.txt. So alive subdomains are as follows.

Kali linux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

crt.sh | %.tesla.com - M... Kazam tharindu

File Actions Edit View Help

tharindu@kali: ~/Desktop x tharindu@kali: ~/Desktop x

```
(tharindu@kali) - [~/Desktop]
$ cat alive.txt
https://www.tesla.com
http://www.tesla.com
https://3.tesla.com
https://akamai-apigateway-crawford-stg.tesla.com
https://akamai-apigateway-crawford-prd.tesla.com
http://akamai-apigateway-crawford-stg.tesla.com
http://akamai-apigateway-crawford-prd.tesla.com
https://akamai-apigateway-dev-warptmsapiserver.tesla.com
https://akamai-apigateway-finplat-prd.tesla.com
http://akamai-apigateway-finplat-prd.tesla.com
http://akamai-apigateway-dev-warptmsapiserver.tesla.com
https://akamai-apigateway-inventorytxnextapi.tesla.com
https://www.tesla.com
http://www.tesla.com
https://akamai-apigateway-einvoicing.tesla.com
http://akamai-apigateway-einvoicing.tesla.com
https://akamai-apigateway-finplat-prd.tesla.com
https://3.tesla.com
http://akamai-apigateway-finplat-prd.tesla.com
http://3.tesla.com
https://akamai-apigateway-captiveunderwriting.tesla.com
https://akamai-apigateway-bender.tesla.com
http://akamai-apigateway-captiveunderwriting.tesla.com
http://akamai-apigateway-bender.tesla.com
https://akamai-apigateway-automation.tesla.com
https://akamai-apigateway-inventorytxnextapi.tesla.com
http://akamai-apigateway-inventorytxnextapi.tesla.com
http://akamai-apigateway-dev-warptmsapiserver.tesla.com
http://akamai-apigateway-deliveryopsapi.tesla.com
https://akamai-apigateway-finplateng.tesla.com
http://akamai-apigateway-finplateng.tesla.com
https://akamai-apigateway-finplateng-routeone.tesla.com
http://akamai-apigateway-finplateng-routeone.tesla.com
https://akamai-apigateway-profileapi.tesla.com
http://akamai-apigateway-profileapi.tesla.com
https://akamai-apigateway-qa-captiveunderwriting.tesla.com
```

## Identifying Website Technologies

We need to figure out the target website's technology platform, such as the content management system and frameworks. These technologies may have vulnerabilities that we may look at during the vulnerability assessment process.

### Webtech

Webtech is a tool that is built with Kali and can be used to figure out website technologies and frameworks. Following is the Webtech scan results for our target.

```
(tharindu@kali) - [~/Desktop]
$ webtech -u https://www.tesla.com/
Wappalyzer Database file not present.
Updating database...
The Wappalyzer database seems offline. Report this issue to: https://github.com/ShielderSec/webtech/
Target URL: https://www.tesla.com/
Detected the following interesting custom headers:
- X-Powered-By: PHP/7.4.16
- X-Drupal-Dynamic-Cache: MISS
- X-UA-Compatible: IE=edge
- X-Generator: Drupal 8 (https://www.drupal.org)
- X-Drupal-Cache: HIT
- X-TZLA-EDGE-HOSTNAME-VCL: drupal8-prod
- X-TZLA-EDGE-BACKEND-FETCH-IF-STALE: true
- X-TZLA-EDGE-WAS-304: false
- X-TZLA-EDGE-Age: 60.000
- X-TZLA-EDGE-Grace: 86400.000
- X-TZLA-EDGE-BACKEND-RETRY: 0
- X-TZLA-EDGE-BACKEND-CONN-TIME: 0.000
- X-TZLA-EDGE-BACKEND-TTFB: 0.000
- X-TZLA-EDGE-BACKEND-REASON: OK
- X-TZLA-EDGE-BACKEND-STATUS: 200
- X-Varnish: 217877074 174129108
- X-TZLA-EDGE-Cache-Hit: Hit
- X-TZLA-EDGE-TTL: 29.842
- X-TZLA-EDGE-GRACE-BACKEND-UNHEALTHY: 86400.000
- X-TZLA-EDGE-BACKEND-STREAM: false
- X-TZLA-EDGE-CLIENT-RESTARTS: 0
- X-TZLA-EDGE-CLIENT-REQ-TTL: -1.000
- X-TZLA-EDGE-Server: sjc38pltegv64.teslamotors.com
- X-TZLA-EDGE-Cache-Hits: 3
- Origin_hostname: drupal8-prod.teslamotors.com, drupal8-prod.teslamotors.com
```



## Builtwith.com

<https://builtwith.com> is a website that may be used to learn about practically all of the technologies that are used in a website. It's quite easy to use; all we must do is supply the target name, which is as follows

### Name Server

[View Global Trends](#)

#### Ireland Domain Redirect

[Ireland Domain Redirect Usage Statistics](#) · [Download List of All Websites using Ireland Domain Redirect](#)

A website having a domain redirect from .ie (Ireland) could be considered a premium business as .ie domain registrations must show either a presence in Ireland, a WIPO registered trademark or significant business from Ireland.

#### Akamai DNS

[Akamai DNS Usage Statistics](#) · [Download List of All Websites using Akamai DNS](#)

DNS services provided by Akamai.

#### UltraDNS neustar

[UltraDNS neustar Usage Statistics](#) · [Download List of All Websites using UltraDNS neustar](#)

DNS services provided by UltraDNS neustar.

### Web Hosting Providers

[View Global Trends](#)

#### Akamai Hosted

[Akamai Hosted Usage Statistics](#) · [Download List of All Websites using Akamai Hosted](#)

Data network CDN provider.

US hosting

## SSL Certificates

[View Global Trends](#)

### SSL by Default

[SSL by Default Usage Statistics](#) · [Download List of All Websites using SSL by Default](#)

The website redirects traffic to an HTTPS/SSL version by default.

### DigiCert SSL

[DigiCert SSL Usage Statistics](#) · [Download List of All Websites using DigiCert SSL](#)

Certificate provided by DigiCert.

Root Authority

### GeoTrust SSL

[GeoTrust SSL Usage Statistics](#) · [Download List of All Websites using GeoTrust SSL](#)

Certificate provided by GeoTrust.

Root Authority

## Web Servers

[View Global Trends](#)

### Varnish

[Varnish Usage Statistics](#) · [Download List of All Websites using Varnish](#)

Varnish is a web accelerator / reverse proxy caching server.

### nginx

[nginx Usage Statistics](#) · [Download List of All Websites using nginx](#)

nginx [engine x] is a HTTP server and mail proxy server written by Igor Sysoev.

### Apache

[Apache Usage Statistics](#) · [Download List of All Websites using Apache](#)

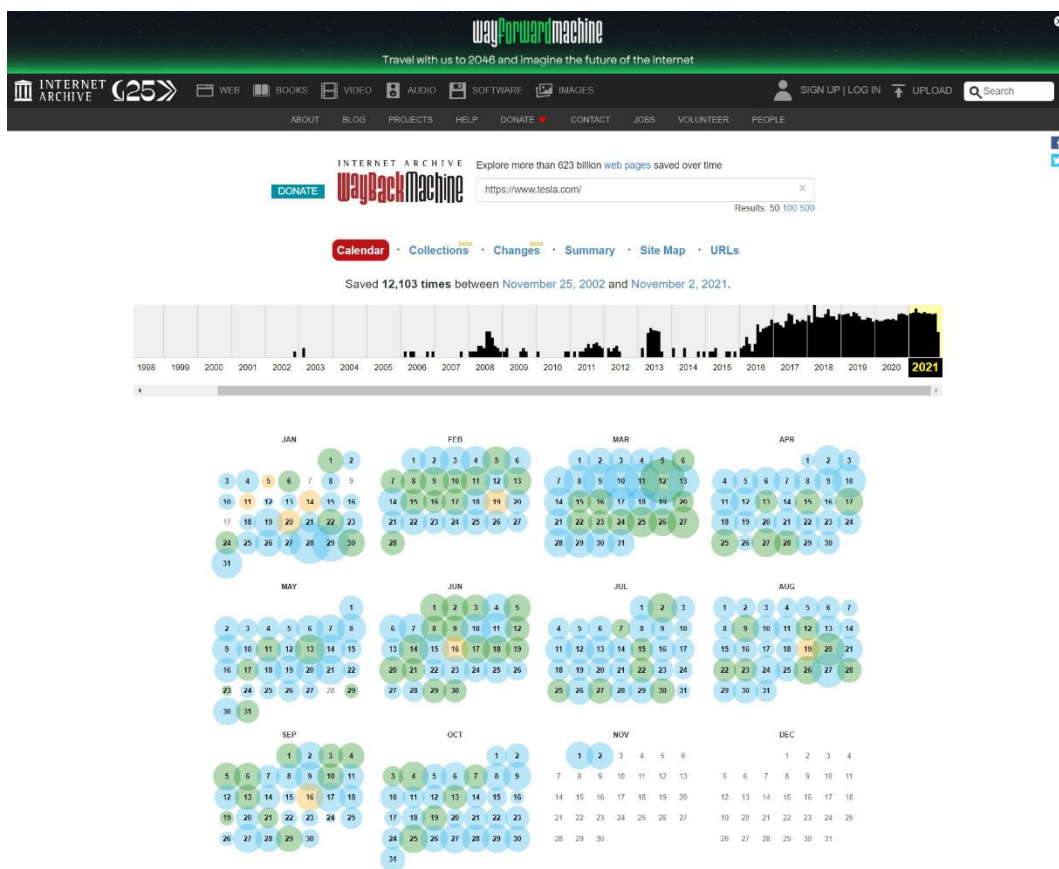
Apache has been the most popular web server on the Internet since April 1996.

Builtwith gives us so many information including their web servers, Name servers, SSL certificates and many more. However, it is more than the Webtech search results.

## Finding Archived Information

### Way Back Machine

Through this, we can see the whole evolution of the target when it was started from scratch. What does it feel like to travel back in time and see how things used to be? The Wayback Machine is here to assist you. It comprises of a massive collection of archived material that we may access online to look back at the history of our website. There is information like as copies of web pages, books, films, audios, and photos that we may examine. It's easy to use; all you must do is supply our target website, which is as follows. Link :<https://web.archive.org/>

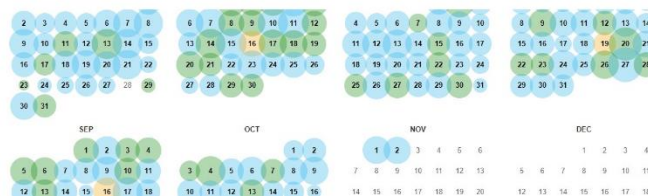


#### Note

This calendar view maps the number of times <https://www.tesla.com/> was crawled by the Wayback Machine, *not* how many times the site was actually updated. More info in the FAQ.

Orange indicates that the URL was not found (4xx).

Green indicates redirects (3xx).

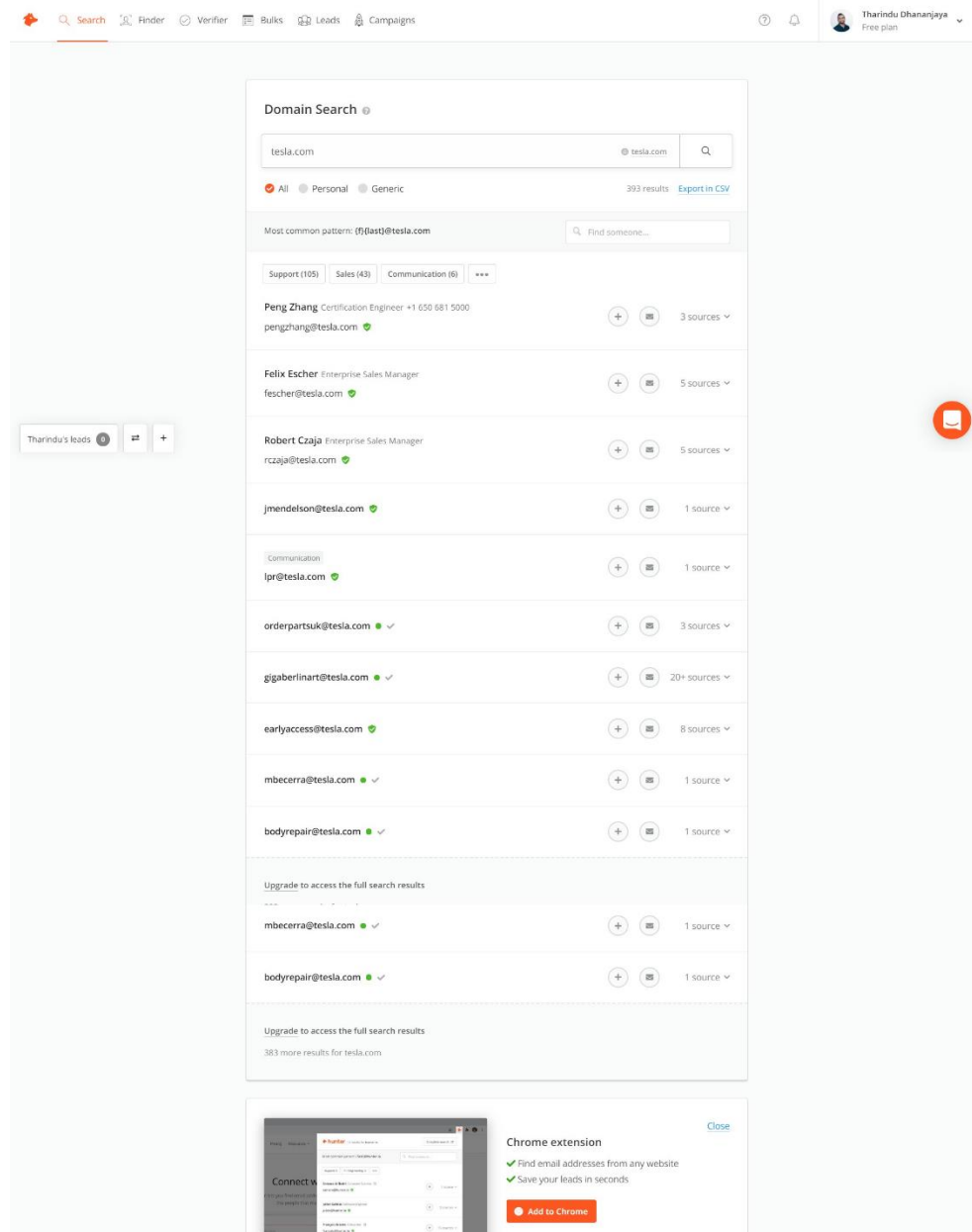


## Email Collecting

We'll use leaked databases on the internet to look for probable emails, usernames, and passwords in this stage. We can utilize brute-force assaults if we can find e-mail and username patterns.

### Hunter.io

<https://hunter.io> is a website that can be used to collect e-mail addresses within an organization. We only need to make an account on their website and search for our target in the way described below.



It includes not only the e-mails, but also the first and last names, as well as the e-mail addresses and the sources from which they came. Most of them came from social networking sites like LinkedIn and Facebook.

We can utilize these e-mails in brute-force attacks and password spraying assaults to get access to the target system as a privileged user. We can also use programs like **Breach-Parse** (<https://github.com/hmaverickadams/breach-parse>) to search for compromised credentials in those e-mails, which scans passwords from a 45GB broken password dump. Although the passwords may have changed by now, some individuals may have changed them in a way that differs somewhat from the prior password. So, we'll give it a go.

## Social media link hunting

**MOSINT** is a Python script that allows us to harvest important information from e-mails such as social profiles, data breaches linked with the e-mail, phone numbers, and related domains, among other things. It's as easy as supplying an e-mail address, but you'll need to git clone it from <https://github.com/alpkeskin/mosint> first. Then, in order to supply your API keys, you must change the config file.

```
(tharindu@kali) - [~/Desktop/mosint]
$ go run main.go -e pengzhang@tesla.com -all

mosint

v2.0
https://github.com/alpkeskin/
Now: Wednesday, 3 Nov 2021

Email > pengzhang@tesla.com
[+] Email verified!
Which Social Media Does pengzhang@tesla.com Use?
[-] Not Found!
Related Emails:
[+]
[+] admin@dnstinations.com
From hunter.io:
[-] Enter the API key in the keys.json file to use this feature!
Related Domains:
[-] Not found!
```

# EmailRep.io API Results:

[ - ] Enter the API key in the keys.json file to use this feature!

## Domain Investigation:

NAME	RECORD
IP	199.66.11.62
NS	edns69.ultradns.com.
NS	a28-65.akam.net.
NS	a7-66.akam.net.
NS	a1-12.akam.net.
NS	a9-67.akam.net.
NS	a10-67.akam.net.
NS	a12-64.akam.net.
MX	tesla-com.mail.protection.outlook.com.

## IPapi.co data:

IP: 199.66.11.62

-- City: New York  
-- Region: New York  
-- Region: New York  
-- Country Name: United States  
-- Country calling code: +1  
-- Timezone: America/New\_York  
-- asn: AS394161  
-- org: TESLA

## Subdomains:

[+] teslacdnpna0.tesla.com  
[+] url7051.tesla.com  
[+] akamai-apigateway-stg-deliveryopsapi1.tesla.com  
[+] akamai-apigateway-deliveryopsapi1.tesla.com  
[+] email1.tesla.com  
[+] vpn1.tesla.com  
[+] sjc04dlrsaap02.tesla.com  
[+] akamai-apigateway-stg-packaging2.tesla.com  
[+] 3.tesla.com  
[+] model3.tesla.com  
[+] url4104.tesla.com

# Footprint and Scanning

## Scanning

A vulnerability is a flaw in a system that, if exploited effectively, can lead to the system being attacked, hence vulnerability management is critical. To control vulnerabilities, however, they must first be detected. It can only be accomplished through a thorough vulnerability scan.

A vulnerability scanner is a program that detects and inventories all systems. The system then compares each of these systems to well-known vulnerabilities and scans for the presence of such flaws in the system.

Finally, the scanner generates a list of systems with any vulnerabilities that need to be addressed noted. The following are a few popular open source scanners:

- Nikto
- Retina
- Wireshark
- OpenVAS

### **Nikto**

Nikto is a vulnerability scanning tool that scans the target web application for harmful files and scripts, as well as other concerns. Nikto performs a wide range of security tests, as well as a scan for out-of-date and unpatched software, as well as harmful files on the web server. Nikto can detect a wide range of problems as well as look for setup errors.



```

(tharindu@kali) - [~]
$ nikto -h
Option host requires an argument

-config+      Use this config file
-Display+     Turn on/off display outputs
-dbcheck      check database and other key files for syntax errors
-Format+      save file (-o) format
-Help         Extended help information
-host+        target host/URL
-id+          Host authentication to use, format is id:pass or id:pass:realm
-list-plugins List all available plugins
-output+      Write output to this file
-nossl        Disables using SSL
-no404        Disables 404 checks
-Plugins+     List of plugins to run (default: ALL)
-port+        Port to use (default 80)
-root+        Prepend root value to all requests, format is /directory
-ssl          Force ssl mode on port
-Tuning+      Scan tuning
-timeout+     Timeout for requests (default 10 seconds)
-update       Update databases and plugins from CIRT.net
-Version      Print plugin and database versions
-vhost+       Virtual host (for Host header)
              + requires a value

Note: This is the short help output. Use -H for full help text.

```

The command "nikto -host" is used to specify the target host, which is the website to be scanned. The domain amazon is used as an example of a target. To execute a basic scan against the target, use "nikto -h https://www.tesla.com/"

The "-p" option can also be used to indicate which ports to investigate. Nikto may scan a single, multiple, or range of ports ranging from 1 to 1000. If no port is given, it will default to scanning port 80. The command "nikto -h https://www.tesla.com/-p 1-1000" searches for a range of ports, whereas "nikto -h https://www.tesla.com/-p80,443" searches for several ports simultaneously.

Following results are the in-scope domain tested through Nikto,

**tesla.cn**

```

(tharindu@kali) - [~]
$ nikto -h tesla.cn
+ Nikto v2.1.6
+-----+
+ Target IP:      211.147.80.201
+ Target Hostname: tesla.cn
+ Target Port:    80
+ Start Time:     2021-11-03 04:52:08 (GMT5.5)
+-----+
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: http://assets.teslastatic.com/index1.html

```

## teslamotors.com

```
(tharindu@kali) ~  
$ nikto -h teslamotors.com  
- Nikto v2.1.6  
-----  
+ Target IP: 172.98.192.37  
+ Target Hostname: teslamotors.com  
+ Target Port: 80  
+ Start Time: 2021-11-03 04:55:26 (GMT5.5)  
-----  
+ Server: nginx  
+ The anti-clickjacking X-Frame-Options header is not present.  
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS  
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type  
|
```

## tesla.services

```
(tharindu@kali) ~  
$ nikto -h tesla.services  
- Nikto v2.1.6  
-----  
+ Target IP: 104.21.7.121  
+ Target Hostname: tesla.services  
+ Target Port: 80  
+ Message: Multiple IP addresses found: 104.21.7.121, 172.67.130.76  
+ Start Time: 2021-11-03 04:56:56 (GMT5.5)  
-----  
+ Server: cloudflare  
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS  
+ Uncommon header 'alt-svc' found, with contents: h3=":443"; ma=86400, h3-29=":443"; ma=86400, h3-28=":443"; ma=86400, h3-27=":443"; ma=86400  
+ Uncommon header 'nel' found, with contents: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}  
+ Uncommon header 'report-to' found, with contents: {"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v3?s=X5V1VDKutlibAkICP3Tm33v8eR3XzSAShsFY5uoB0ffU6tuGqQ0RPQ3223LaQt8nUYj8%2BKYZYVvp1vQtCQ0LwxdB8ozjphI2PpqrUnrWqz2Fa60sTFTbEnSalkppI8hiw%3D"}],"group":"cf-nel","max_age":604800}  
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type  
|
```

## solarcity.com

```
(tharindu@kali) ~  
$ nikto -h solarcity.com  
- Nikto v2.1.6  
-----  
+ Target IP: 209.11.133.123  
+ Target Hostname: solarcity.com  
+ Target Port: 80  
+ Start Time: 2021-11-03 05:00:40 (GMT5.5)  
-----  
+ Server: LB  
+ The anti-clickjacking X-Frame-Options header is not present.  
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS  
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type  
+ Root page / redirects to: https://solarcity.com/  
|
```

## Foot printing

The goal of fingerprinting, also known as foot printing, is to gather as much information about a particular online application as possible. It is a collection of data that may be

used to identify network protocols, operating systems, hardware, and software, among other things.

This method is used to authenticate users as a security precaution, however attackers might use it to find weaknesses in the target web application. This contains details such as operating system kinds and versions, domain names, network blocks, VPN points, and so on.

A bespoke packet set is launched to acquire information, and when they obtain a response from the target, information about operating systems, protocols, and other topics is withdrawn.

Passive and active fingerprinting are the two major methods of fingerprinting. Sending packets to the target and waiting for a response, then evaluating the result, is what active fingerprinting entails. Passive fingerprinting is the process of monitoring a target's network data without interfering with it directly.

Tools that could be used to do fingerprinting are;

- Nmap
- Ettercap
- PacketFence
- Netcat

### **Enumerating Open ports with Nmap**

Nmap, commonly known as Network Mapper, is a tool for fingerprinting active stacks. It detects what devices are operating on the targeted system, discovers accessible hosts, finds open ports, and detects security threats, among other things.

Nmap can scan a single host as well as a network with many devices and subnets. Nmap collects data by delivering raw data packets to system ports and determining if the postings are open or closed, or if they are filtered by a firewall or something else.

```
(tharindu@kali) - [~/Desktop/mosint]
$ sudo nmap -sS tesla.com
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-03 06:26 IST
Nmap scan report for tesla.com (199.66.11.62)
Host is up (0.034s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 22.24 seconds

(tharindu@kali) - [~/Desktop/mosint]
$
```

It indicates 3 ports which are 25,80 and 443. Those are the most common ports we can see in every web application. If you need furthermore details, we can run **\$sudo nmap -A -p25,80,443 tesla.com**.

## Brute Forcing Directories

In each web program, there are hidden or inaccessible folders that provide useful information for a penetration tester. These folders can be found using open-source technologies that are freely accessible. We'll need to employ a variety of tools for this because each one has its own set of benefits and drawbacks.

## Fingerprinting Web Application Firewall with WAFW00F

WAFW00F is a python-based penetration testing tool that comes pre-installed in penetration testing deployments. If it isn't already installed, you may git clone it from <https://github.com/EnableSecurity/wafw00f>. It may fingerprint a web application firewall by sending specially constructed web requests to the server and evaluating the answers.

```
(tharindu@kali)-[~/wafw00f]
$ wafw00f http://tesla.com
```

~ WAFW00F : v2.1.0 ~  
The Web Application Firewall Fingerprinting Toolkit

```
[*] Checking http://tesla.com  
[+] The site http://tesla.com is behind CacheWall (Varnish) WAF.  
[~] Number of requests: 2
```

We can see the website is behind CacheWall firewall.

## Reconnaissance

In order to acquire additional information and do footprinting scanning, the Ghost Eye automated python script was employed. It's available on GitHub at [https://github.com/BullsEye0/ghost\\_eye](https://github.com/BullsEye0/ghost_eye). The following is an example of how to use it. You must indicate the tasks that must be completed.

[illegible]

The following is a Whois Lookup for our target domain. Whois Lookup offers you a general notion of our target's registration details. It occasionally provides useful information, such as the e-mail addresses of website administrators.

```
Erase is control-H (^H).
[~] Searching for Whois Lookup: tesla.com
Domain Name: TESLA.COM
Registry Domain ID: 187902_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2020-10-02T09:07:57Z
Creation Date: 1992-11-04T05:00:00Z
Registry Expiry Date: 2022-11-03T05:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: A1-12.AKAM.NET
Name Server: A10-67.AKAM.NET
Name Server: A12-64.AKAM.NET
Name Server: A28-65.AKAM.NET
Name Server: A7-66.AKAM.NET
Name Server: A9-67.AKAM.NET
Name Server: EDNS69.ULTRADNS.BIZ
Name Server: EDNS69.ULTRADNS.COM
Name Server: EDNS69.ULTRADNS.NET
Name Server: EDNS69.ULTRADNS.ORG
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2021-11-02T06:26:18Z <<<
```

```
ERASE IS CONTROL-H (^H).
[~] Searching IP Location Finder: tesla.com

[+] Url: tesla.com
[+] IP: 199.66.11.62
[+] Status: success
[+] Region: New York
[+] Country: United States
[+] City: New York
[+] ISP: Tesla, Inc.
[+] Lat & Lon: 40.7128 & -74.006
[+] Zipcode: 10123
[+] TimeZone: America/New_York
[+] AS: AS394161 Tesla, Inc.
```

Many useful pieces of information were discovered,

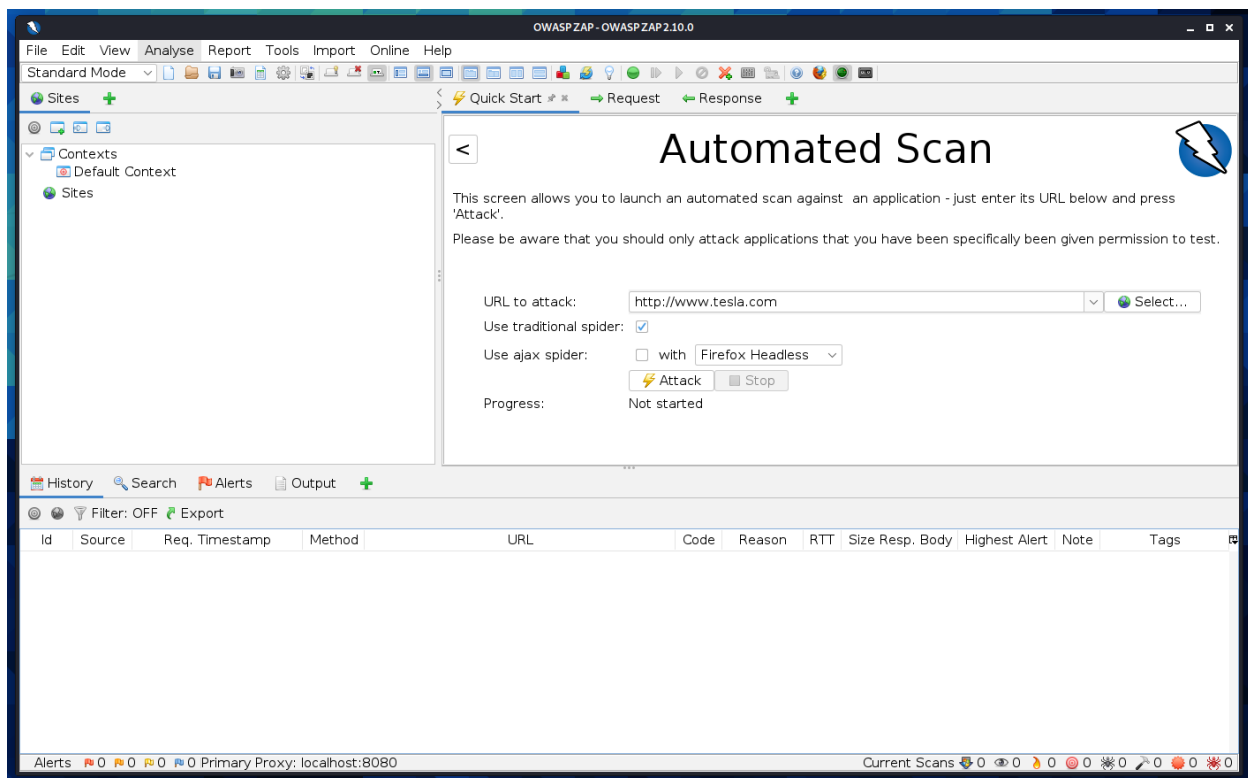
including nation, area, internet service provider, and IP address. If our assessment

scope allows for social engineering assaults, this type of knowledge comes in useful, since we can use it to launch effective attacks. However, I could not find any valuable information.

## Owasp ZAP

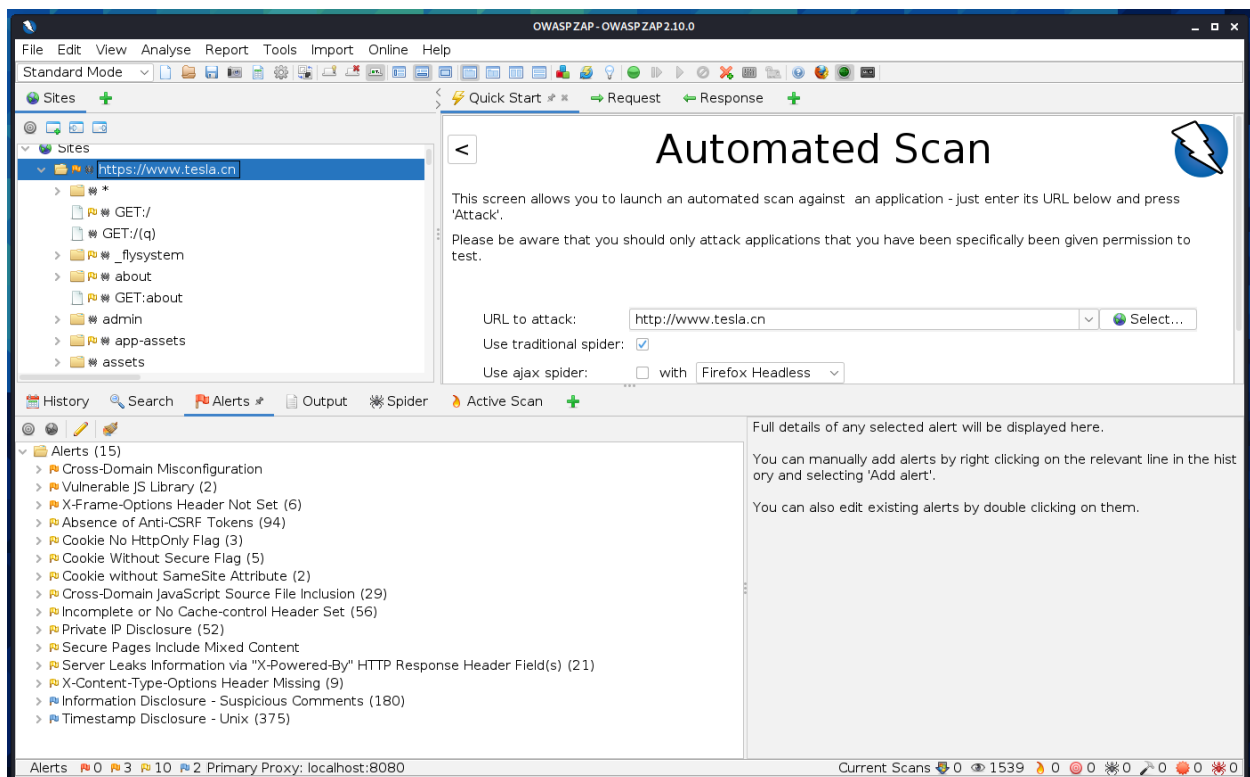
The open-source web application security scanner OWASP ZAP (short for Zed Attack Proxy) was developed by OWASP. When used as an intermediate worker, it gives the client complete control over all traffic that passes through it, including https traffic. It can also operate in daemon mode, which can then be controlled through a REST API.

Intercepting intermediate worker, Traditional and AJAX Web crawlers, Automated scanner, Passive scanner, Forced perusing, Fuzzer, WebSocket support, Scripting languages, and Plug-n-Hack support are some of the features of the scanner. It contains a module-based engineering system and an online 'commercial center,' which allows for the addition of new or updated highlights. The graphical user interface (GUI) control panel is not difficult to operate. Let's choose the automate scan.



Here are the results of in-scope domains.





# Vulnerability Assessment

This phase focuses on compiling a list of vulnerabilities found on the target systems and categorizing them according on the risk they pose. Each target discovered in the preceding phases must be subjected to a vulnerability evaluation by a penetration tester. Because the exploitation phase will go over this list of vulnerabilities, a longer and more thorough list will give you a better chance of exploiting the systems. It's important for testers to remember to look for any and all vulnerabilities in the target system.

There are two ways to carry out a vulnerability assessment,

1. Manually — using the previously collected data.

## 2. Utilizing automated tools

During this phase, a skilled penetration tester will typically employ both automated tools and manual inspection. During a penetration test, testers should keep in mind that automated tools can assist them. They do not, however, conduct a penetration test on their own. We must manually verify the reliability of automatic vulnerability assessment tool scan results. As a result, this assessment is carried out using a combination of manual inspection and automated technologies, with the targets being assessed against the OWASP top 10 as well as other vulnerabilities.

### Netsparker and Nikto

Such attacks are launched by exploiting a weakness in the system itself. As a result, such weaknesses must be identified and patched as part of a regular security audit.

All of the chosen subdomains were examined for vulnerabilities using Netsparker and nikto.

[www.teslamotors.com](http://www.teslamotors.com)

### Missing X-Frame-Options Header

LOW

Certainty :   
URL : <http://www.teslamotors.com/>

### Vulnerability Details

Netsparker detected a missing `X-Frame-Options` header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options` HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a `frame` or an `iframe`. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

## Impact

Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.

## Remedy

- Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
  - X-Frame-Options: DENY It completely denies to be loaded in frame/iframe.
  - X-Frame-Options: SAMEORIGIN It allows only if the site which wants to load has a same origin.
  - X-Frame-Options: ALLOW-FROM URL It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
- Employing defensive code in the UI to ensure that the current frame is the most top level window.

## Missing Content-Type Header

LOW

Certainty : 

URL : [https://www.teslamotors.com/?js=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJKb2t1bpc3MiOiJKb2t1b1IsImpzIjoxLCJqdGkiOiIycXEyZ2oxMmVxMzFxbjUycG8wazVoa2kiLCJuYmYiOiJE2MzU5NTtiPpN9pq2T\\_bxcqH9ZXwZpKtvz6tMY&sid=73989cc2-3cf1-11ec-ae98-e3b000745cee](https://www.teslamotors.com/?js=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJKb2t1bpc3MiOiJKb2t1b1IsImpzIjoxLCJqdGkiOiIycXEyZ2oxMmVxMzFxbjUycG8wazVoa2kiLCJuYmYiOiJE2MzU5NTtiPpN9pq2T_bxcqH9ZXwZpKtvz6tMY&sid=73989cc2-3cf1-11ec-ae98-e3b000745cee)

## Vulnerability Details

Netsparker detected a missing Content-Type header which means that this website could be at risk of a MIME-sniffing attacks.

## Impact

MIME type sniffing is a standard functionality in browsers to find an appropriate way to render data where the HTTP headers sent by the server are either inconclusive or missing.

This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the intended content type.

The problem arises once a website allows users to upload content which is then published on the web server. If an attacker can carry out XSS (Cross-site Scripting) attack by manipulating the content in a way to be accepted by the web application and rendered as HTML by the browser, it is possible to inject code in e.g. an image file and make the victim execute it by viewing the image.

## Remedy

1. When serving resources, make sure you send the content-type header to appropriately match the type of the resource being served. For example, if you are serving an HTML page, you should send the HTTP header:

Content-Type: text/html

2. Add the X-Content-Type-Options header with a value of "nosniff" to inform the browser to trust what the site has sent is the appropriate content-type, and to not attempt "sniffing" the real content-type.

X-Content-Type-Options: nosniff

## SSL/TLS Not Implemented

MEDIUM

URL : <https://www.teslamotors.com/>

### Vulnerability Details

Netsparker detected that SSL/TLS is not implemented.

## Impact

An attacker who is able to intercept your - or your users' - network traffic can read and modify any messages that are exchanged with your server.

That means that an attacker can see passwords in clear text, modify the appearance of your website, redirect the user to other web pages or steal session information.

Therefore no message you send to the server remains confidential.

### Remedy

We suggest that you implement SSL/TLS properly, for example by using [the Certbot tool](#) provided by the Let's Encrypt certificate authority. It can automatically configure most modern web servers, e.g. Apache and Nginx to use SSL/TLS. Both the tool and the certificates are free and are usually installed within minutes.

## User Controllable Cookie

CONFIRMED

LOW

URL : <http://www.teslamotors.com/?js=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ2ODI2LCJpc3MiOiJKb2tlbiIsImpzIjoxLCJqdGkiOiIycXEyZ2djYWYwMGJka2dyIiwfQ.oikcMUwEAPZY9hkM7Cke99Kh6-av8nsZ0vmV9vijf0Y&sid=ns:netsparker>

Identified Cookie(s) : **sid**

Cookie Source : HTTP Header

Parameter Name : sid

Parameter Type : GET

Attack Pattern : ns%3anetsparker056650%3dvuln

### Vulnerability Details

Netsparker identified a user controllable cookie.

## Impact

Attackers can easily set an arbitrary value in the cookie and this may allow them to bypass authentication, carry out attacks such as SQL injection and cross-site scripting or modify inputs in unexpected ways.

## Remedy


Add integrity checks and server side validation to detect tampering.

Service.tesla.com

# [Possible] Internal IP Address Disclosure

LOW

---

Certainty : 

URL : <https://service.tesla.com/styles.6d36f891b11fb1586e03.css>

Extracted IP Address(es) :   
10.24.74.74  
10.24.72.72

ExtractedIPAddresses :   
10.24.74.74  
10.24.72.72

## Vulnerability Details

Netsparker identified a Possible Internal IP Address Disclosure in the page.

It was not determined if the IP address was that of the system itself or that of an internal network.

## Impact

There is no direct impact; however, this information can help an attacker identify other vulnerabilities or help during the exploitation of other identified vulnerabilities.

## Remedy

First, ensure this is not a false positive. Due to the nature of the issue, Netsparker could not confirm that this IP address was actually the real internal IP address of the target web server or internal network. If it is, consider removing it.

## HTTP Strict Transport Security (HSTS) Errors and Warnings

MEDIUM

Certainty :   
URL : <https://service.tesla.com/>

### CLASSIFICATION

OWASP 2013	<a href="#">A5</a>
OWASP 2017	<a href="#">A6</a>
CWE	<a href="#">16</a>
WASC	<a href="#">15</a>
ISO27001	<a href="#">A.14.1.2</a>

Error	Resolution
preload directive not present	Submit domain for inclusion in browsers' HTTP Strict Transport Security (HSTS) preload list.

## Vulnerability Details

Netsparker detected errors during parsing of Strict-Transport-Security header.



## Impact

The HSTS Warning and Error may allow attackers to bypass HSTS, effectively allowing them to read and modify your communication with the website.

## Remedy

Ideally, after fixing the errors and warnings, you should consider adding your domain to the the HSTS preload list. This will ensure that browsers automatically connect your website by using HTTPS, actively preventing users from visiting your site using HTTP. Since this list is hardcoded in users' browsers, it will enable HSTS even before they visit your page for the first time, eliminating the need for Trust On First Use (TOFU) with its associated risks and disadvantages. Unless you fix the errors and warnings your website won't meet the conditions required to enter the browser's preload list.

Browser vendors declared:

- Serve a valid certificate
- If you are listening on port 80, redirect all domains from HTTP to HTTPS on the same host. Serve all subdomains over HTTPS:
  - In particular, you must support HTTPS for the www subdomain if a DNS record for that subdomain exists
- Serve an HSTS header on the base domain for HTTPS requests:
  - The `max-age` must be at least 31536000 seconds (1 year)
  - The `includeSubDomains` directive must be specified
  - The `preload` directive must be specified
  - If you are serving an additional redirect from your HTTPS site, that redirect must have the HSTS header (rather than the page it redirects to)

## Cookie Not Marked as HttpOnly

CONFIRMED

LOW

URL : <https://service.tesla.com/>

Identified Cookie(s) : `lang`  
`returnURL`

Cookie Source : JavaScript

## Vulnerability Details

Netsparker identified a cookie not marked as HTTPOnly.

HTTPOnly cookies cannot be read by client-side scripts, therefore marking a cookie as HTTPOnly can provide an additional layer of protection against cross-site scripting attacks.

## Impact

During a cross-site scripting attack, an attacker might easily access cookies and hijack the victim's session.

## Actions to Take

1. See the remedy for solution.
2. Consider marking all of the cookies used by the application as HTTPOnly. (*After these changes javascript code will not be able to read cookies.*)

## Remedy

Mark the cookie as HTTPOnly. This will be an extra layer of defense against XSS. However this is not a silver bullet and will not protect the system against cross-site scripting attacks. An attacker can use a tool such as [XSS Tunnel](#) to bypass HTTPOnly protection.

## Solarcity.com

```
(tharindu@kali) ~/Desktop/mosint
$ nikto -h solarcity.com
- Nikto v2.1.6

-----
+ Target IP:      209.11.133.123
+ Target Hostname: solarcity.com
+ Target Port:    80
+ Start Time:     2021-11-04 04:11:00 (GMT5.5)
-----
+ Server: LB
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: https://solarcity.com/
```

## Anti-clickjacking X Frame Option header is not present

**Clickjacking :** Clickjacking is a harmful technique for tricking users into clicking on an object that is disguised as the thing the user expects to click on. The hidden page/item

can lead to people visiting dangerous websites, downloading malware, or visiting a genuine website they didn't plan to visit.

Because the user believes they are on the site they wanted to visit, they can disclose sensitive information, transfer money, and buy items, among other things. Clickjacking attacks such as likejacking and cursorjacking are well-known.

The x-frame option header has three values that tell the browser whether to allow or restrict framing from other domains:

- DENY – prevents the page from being displayed on any domain
- SAMEORIGIN – enables the page to be displayed on another page but only inside the current domain
- ALLOW-FROM URI – allows the page to be displayed in a frame but only in that frame

## Insecure Transportation Security Protocol Supported (TLS 1.0)

CONFIRMED

LOW

URL : <https://solarcity.com/>

### Vulnerability Details

Netsparker detected that insecure transportation security protocol (TLS 1.0) is supported by your web server.

TLS 1.0 has several flaws. An attacker can cause connection failures and they can trigger the use of TLS 1.0 to exploit vulnerabilities like BEAST (Browser Exploit Against SSL/TLS).

Websites using TLS 1.0 are considered non-compliant by PCI since 30 June 2018.

## Impact

Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.

## Actions to Take

We recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher. See Remedy section for more details.

## Remedy

Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod\_ssl module. This directive can be set either at the server level or in a virtual host configuration.
  - `SSLProtocol +TLSv1.2`
- For Nginx, locate any use of the directive `ssl_protocols` in the `nginx.conf` file and remove `TLSv1`.
  - `ssl_protocols TLSv1.2;`
- For Microsoft IIS, you should make some changes on the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**
  1. Click on Start and then Run, type `regedt32` or `regedit`, and then click OK.
  2. In Registry Editor, locate the following registry key or create if it does not exist:
    3. `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\`
  4. Locate a key named `Server` or create if it doesn't exist.
  5. Under the `Server` key, locate a DWORD value named `Enabled` or create if it doesn't exist and set its value to "0".
- For lighttpd, put the following lines in your configuration file:
  - `ssl.use-ssl2 = "disable"`
  - `ssl.use-ssl3 = "disable"`
  - `ssl.openssl.ssl-conf-cmd = ("Protocol" => "-TLSv1.1, -TLSv1, -SSLv3") # v1.4.48 or up`
  - `ssl.ec-curve = "secp384r1"`

# Conclusion

Overall, the audited web application was well-designed and has several security features in place to protect itself from cyber-attacks. There were only medium to low risk vulnerabilities discovered throughout the testing, as well as best practices, and no high-risk vulnerabilities, which is a solid indicator that a website is safe and effectively protected.

All of the procedures and instruments utilized in the evaluation, as well as the results, are detailed. Furthermore, the effect of the discovered vulnerabilities, verification of their presence, and steps to take to reduce the risk associated with the vulnerabilities are all detailed.

When examining the domain, no high-risk vulnerabilities were discovered, however medium-risk and low-risk vulnerabilities were discovered. As a result, the target domain's total risk is medium. All the vulnerabilities are categorized according to the risk level, and the OWASP Top 10 category.

# References

<https://resources.infosecinstitute.com/topic/ssl-attacks/>

<https://github.com/vavkamil/awesome-bugbounty-tools>

<https://resources.infosecinstitute.com/topic/14-popular-web-application-vulnerability-scanners/>

<https://kalilinuxtutorials.com/wafw00f/>

[https://www.youtube.com/watch?v=IZAoFs75\\_cs&list=RDCMUC8butISFwT-WI7EV0hUK0BQ&index=2](https://www.youtube.com/watch?v=IZAoFs75_cs&list=RDCMUC8butISFwT-WI7EV0hUK0BQ&index=2)

<https://www.youtube.com/watch?v=2nXOxLpeu80&list=RDCMUC8butISFwT-WI7EV0hUK0BQ&index=3>