

# Crafts Team Security Report



**Oregon State**  
University

February 2023

Prepared by:

Dam Security



## Table of Contents

|                               |    |
|-------------------------------|----|
| 1. Cover Page                 | 1  |
| 2. Table of Contents          | 2  |
| 3. Report Introduction        | 3  |
| 4. Crafts Team Background     | 4  |
| 5. OSU Security Policies      | 7  |
| 6. Vulnerabilities            | 10 |
| 6.1. Threat model             |    |
| 6.2. CVE                      |    |
| 6.3. Tools                    |    |
| 7. Recommendations            | 11 |
| 7.1. Overview of security     |    |
| 7.2. Getting to the next tier |    |
| 7.3. Documenting attacks?     |    |
| 8. Conclusion                 | 12 |
| 9. Research Process           | 14 |



## Report Introduction

Cyber security policies are essential for colleges to ensure the safekeeping of student information and protect the security of their networks and systems. With an increasing number of students connecting to college networks and needing access to various digital services, the danger of cyber-attacks and data breaches makes it necessary to have an effective cybersecurity policy in place. A cyber security policy not only provides a sense of protection for the college, but also provides guidelines for students, faculty, and staff on how to use technology safely and securely. Additionally, having a strong cyber security policy can help to protect the college from legal liabilities and reputational damage in the case of a data breach. Ultimately, cyber security policies are important for colleges in order to provide a secure and safe environment for students and staff.

### This Report will outline

1. OSU Security Policies and how to ensure your site is safe
2. Crafts Team security vulnerabilities
3. How to protect the site??

The scope of this report is to provide a comprehensive overview of cyber security threats and vulnerabilities. It will analyze current trends, identify potential security issues, and outline best practices to mitigate risks.

Cyber security threats come in many forms, from malware and viruses to phishing and DDoS attacks. Hackers use sophisticated methods to gain access to personal data, financial accounts, and other confidential information stored online. Cybercriminals can also exploit vulnerable systems and networks to steal data and infect computers with malicious software. In addition, wireless networks and the Internet of Things (IoT) have increased the attack surface, allowing attackers to access many devices, networks, and systems in one go.

It is important to identify potential security vulnerabilities, such as weak passwords, lack of encryption, and insecure software/hardware configurations. Companies should also have a robust security policy, regularly update their systems, and train staff on security awareness. Additionally, businesses should conduct regular penetration testing and employ security solutions such as firewalls, anti-malware, and anti-virus software.

Finally, the report should provide recommendations on how to respond to and mitigate cyber security threats. This could include steps to assess, detect, and respond to threats, in addition to measures to prevent and reduce the impact of any attack. By following the best practices outlined in this report, companies can ensure their networks and data are secure and protected from cyber threats.

## Crafts Team Background

### What We Do

The OSU Craft Center is your Student Campus Creative Resource. We are the place to nurture your inner artist and nourish your soul. The Craft Center offers well-equipped studios and classes in the following areas: Ceramics, Glass, Woodworking, Fibers, Pen & Paper, and Jewelry/Metals. Membership is open to OSU students. Membership is FREE to incidental fee-paying students and fee-based for other OSU students (see our Front Desk staff for more information). All levels are welcome from beginner to advanced. Classes are taught by skilled artisans, with an emphasis on quality small group instruction and individual attention. Come learn, explore and create in a warm and friendly environment!

Student groups, clubs, and orgs. and departments can partner with us to leverage our expertise and technical support to enhance programming that aligns with our shared goals.

### Our Mission & Vision

To support and promote a creative outlet of expression and an enriched experiential learning experience where students can foster their creativity and skill sets. The Craft Center provides opportunities for OSU Students to engage their hearts and minds in the pursuit of self-discovery and self-expression. We provide pathways for student leaders to discover what is important to them, as they build upon their skills and confidence through meaningful service to others. We are committed to having the OSU Craft Center be a welcoming, inclusive and equitable creative resource for all OSU Students.

With well-equipped studios and an extensive series of workshops and classes in the handcrafted/visual arts, the Craft Center complements and augments the educational opportunities available at Oregon State University.

### Goals

- **Experiential Learning:** We create enriching hands-on learning opportunities
- **Wellness:** We provide an environment and activities that foster the well-being of students
- **Leadership:** We create opportunities for students to learn, lead and teach others
- **Community:** We provide an engaging, welcoming and inclusive environment where creative thinkers can find community.
- **Collaboration:** We design programming specifically to meet the needs of a diverse campus; whether championing social causes, celebrating cultural diversity, building community and a climate of acceptance, valuing community service and giving back.

## **Security Rules of the Office of Information Security**

The Office of Information Security utilizes six primary security rules in order to effectively create a safe, respectful, and ethical online environment.

### **VULNERABILITY MANAGEMENT RULE**

Ensures the assessment of university IT systems in order to determine security vulnerabilities in need of fixing. An essential process for the better protection of university systems and data. This rule applies to all academic, research, and administrative departments and offices at all University locations; all University faculty, staff, students, visitors, contractors and affiliates; and all resources, systems, infrastructure, devices, facilities and applications in the University's computing portfolio, whether located on University property or accessed remotely.

### **APPROPRIATE USE FOR SYSTEM ADMINISTRATORS RULE**

System Administrators manage, configure, monitor and access University Information Resources. This high level of access is a position of trust within the University. Individuals who are granted elevated access are personally responsible for their actions. This Rule establishes Acceptable Use for System Administrators for Oregon State University. This rule establishes requirements for System Administrators to ensure that their elevated level of access is performed in a professional and ethical manner.

### **LOG MANAGEMENT RULE**

Governs the University's current log collection, analysis, and retention methods. Ensuring that all processes involving log management satisfy ethical, contractual, and risk-based requirements. This rule applies to any University department or individual that uses or operates IT resources that support official University business.

### **REMOTE ACCESS RULE**

Defines how Oregon State University controls remote access to University information systems, networks, and resources in order to prevent unauthorized use and to ensure proper use. This rule applies to all users associated with Oregon State University who need to access University resources from the internet.

### **PASSWORD MANAGEMENT RULE**

Outlines the principles and practices of operation for the University's password authentication services. This rule applies to all individuals who use or operate any University system or resource that requires password authentication

### **UIT EMAIL SECURITY RULE**

Outlines the principles and practices of operation for the University's Email Services. This rule applies to any University department or individual that uses or operates an Email Service that supports official University business.

## Vulnerabilities

### 6.1 Threat Model

Identification of any possible threats to our client's software available to the public. The threat modeling process will include identifying threats at each level of the web application from users to the backend to cloud services. Proper threat modeling can help identify countermeasures for each attack surface present in a web application environment. The focus of the threat modeling process on the craft team's project will be to identify the most probable threats and attacks to this web application. Below is a list illustrating expected threats at each level starting with users accessing the web application.

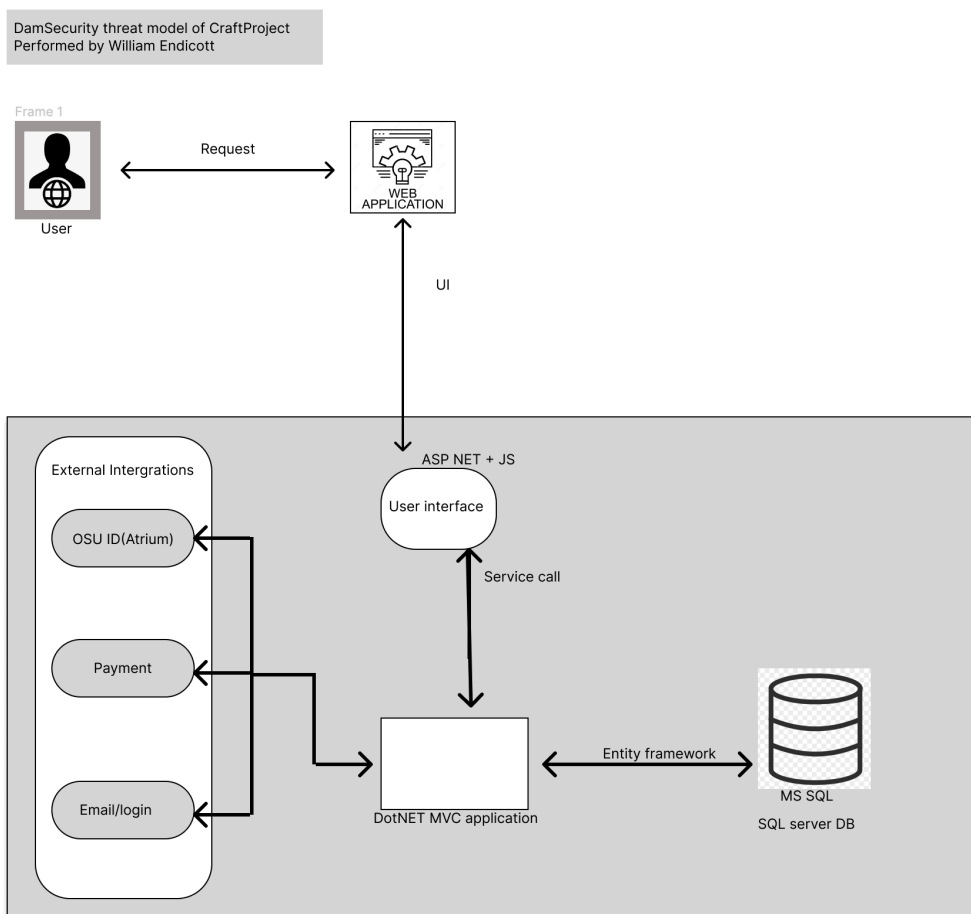


Figure 1. Basic outline of the architecture for the web application, used from the presented documentation and schematic designed by the development team of the craft team's project.

#### Users

Social engineering remains one of the most damaging aspects to businesses and companies. Simple social engineering can lead to malware, ransomware, leaking of credentials, and loss of digital assets. Given the importance of protecting user and employee data, it is necessary to at least minimize phishing attempts via email, phone, and even physical mail. On OWASP's top ten list of security risks, the second vulnerability listed is cryptographic failures. Cryptographic failures can lead to insecure communication. The craft team's project will be



dealing with the use of users' credit cards, banking information, and shipping addresses. The use of this personal information of the users can lead to a man-in-the-middle attack and subsequently a social engineering attack. Assets at risk in a social engineering attack on the craft team's project include customer data, employee data, availability of service, and payment information.

### **Web Application**

The third listed vulnerability on OWASP's top ten is injections. SQL injection (SQLi) and cross-site scripting (XSS) are the most dangerous and common weaknesses in online applications. DoS (denial of service) attacks and attacks involving authentication methods are also common threats. If a website has weak authentication recommendations for its users, that can lead to a higher-than-normal amount of accounts being compromised.

Identification and authentication failures are listed as seventh on the OWASP's top ten. Insecure communication and platform misconfiguration will also lead to various other attacks. Security misconfiguration is the fifth vulnerability listed on the OWASP's top ten. Having all the security in the world will be useless if the application is misconfigured and vulnerable ports remain open. Assets at risk in an attack on the craft team's project involving the web application include user and employee data, user and employee authentication credentials, and sensitive material that may be stored, including payment information and associated addresses.

### **Files/Database**

Files will have similar threats that are included in the database section. Improper encryption of data while at rest and in transit is the greatest concern for this section. A great majority of the threats from the web application section also apply here as well including SQLi and improper platform configuration. The information provided by users will be stored in the database so unencrypted sensitive data at rest needs to be accounted for since that is a commonly missed aspect of security. Files storing employee credentials, admin credentials, and customer data are the assets at risk.

### **Logs**

Logs are an important part of running a service online. Maintaining proper security logs and event logs will help later identify if someone did something malicious on the website. It would be important to log and maintain a running file of documents that are updated and by who as well as any information that can be obtained from the interaction. If someone uploads a malicious file and an offensive file we would want the ability to find that person and hold them responsible. This is where one of the key pillars of information security comes into play. Non-repudiation will allow the administrators of the website to log the sender's information that is provided from the network traffic as well as the credentials associated with the traffic.

## External Integrations

This will handle the login credentials for people accessing the product. Insecure communication is a priority to identify. Risk of attack methods on two-factor authentication are seen at this level. Those include but are not limited to SMS-based man-in-the-middle attacks, pass-the-cookie attacks, and social engineering attacks from the use of 2FA. Overall, this is a service that will be handled outside of the project team's hands but it will need to be checked for the correct configuration.

## 6.2 CVE's

This section will be used to focus on real world examples with common vulnerabilities and exposures(CVE) that are identified on other websites registration pages. This will provide guidance and be used as a reference when inspecting the Craft's team project. The examples will provide direct comparison and something to use as guidance to help improve the Craft team to the next tier of security.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-43097>

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-0232>

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-4073>

[registrationmagic](#)

## 6.3 Tool's

The tools used to do basic reconnaissance and identification of possible vulnerabilities will be discussed in this section. We used three tools during the process of evaluating this project. The tools we used include NMAP, tcpdump, wireshark, and Sn1per. Each one of these tools is used for different purposes but they are closely related in a sense that they're for reconnaissance and less intrusive than other tools available. The order in which they will be covered is NMAP, wireshark, and then finally Sn1per.

### Reconnaissance

Reconnaissance is one of the first five phases listed on the ec-council's website for phases of penetration testing. They describe this phase as gathering as much information about a target system as possible. This will include network topology, the operating system the target is on, applications, and user accounts. As someone new to ethical hacking it's easy to see the importance of this step. Afterall, it is better to know the target to figure out what tools we can use on the target's system.

### Passive Reconnaissance

The ec-council talks about the two categories of reconnaissance. The first is passive and this could consist of publicly available information about a target. This could include gathering domain names, email addresses, IP addresses, and much more. In an article written by Shimon Brathwaite, titled "Active vs Passive Cyber Reconnaissance in Information Security." Brathwaite gives passive reconnaissance a familiar name I have heard before. OSINT is short for open-source intelligence. Brathwaite lists common tools used in the process of gaining

intelligence on a target. The three listed include Google Hacking, Netcraft, and Shodan but many more are available.

### **Active Reconnaissance**

Active reconnaissance requires interacting with the target to gain information. This will require a set of tools to scan a network to find out information. NMAP which is short for network mapper is used to scan systems giving various information on a target. NMAP is a focus for this section and the tool we used to scan target systems and will be discussed in detail shortly. Active reconnaissance has the chance of being detected by the target.

The information gathered at this part of reconnaissance includes but is not limited to finding out if a port is opened or closed, the operating system the machine is using, the services the target is running, and discovering if the host has vulnerable applications or ports. Since this is intrusive it is vital as mentioned before not to scan anything we do not own or have permission to scan. Scanning a network can trigger intrusion detection systems and intrusion prevention systems.

#### **NMAP (network mapping)**

NMAP is the first tool introduced for our journey of reconnaissance. This tool has many features and capabilities, few of which will be discussed here. The first part of understanding nmap will begin with scanning [scanme.nmap.org](https://scanme.nmap.org) checking for its operating system, version, script, and traceroute. In figure 1 below the results of our first scan gave me great details about the target. This gives a great overview of the target with many details that can be a little overwhelming at first glance.

```

(root@kali)~[~]
# nmap -A -T4 scanme.nmap.org
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-06 13:58 PDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.0085s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 992 closed tcp ports (reset)
PORT      STATE      SERVICE      VERSION
22/tcp    open      ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256  96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256  33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
25/tcp    filtered  smtp
80/tcp    open      http         Apache httpd 2.4.7 ((Ubuntu))
|_ http-title: Go ahead and ScanMe!
|_ http-favicon: Nmap Project
|_ http-server-header: Apache/2.4.7 (Ubuntu)
135/tcp   filtered  msrpc
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds
9929/tcp  open      nping-echo   Nping echo
31337/tcp open      tcpwrapped
Aggressive OS guesses: Linux 3.2 (94%), Linux 4.4 (94%), DD-WRT v24-sp2 (Linux 2.4.37) (92%), Ac
tiontec MI424WR-GEN3I WAP (92%), Microsoft Windows XP SP3 or Windows 7 or Windows Server 2012 (9
0%), Microsoft Windows XP SP3 (90%), BlueArc Titan 2100 NAS device (87%), TiVo series 1 (Sony SV
R-2000 or Philips HDR112) (Linux 2.1.24-TiVo-2.5, PowerPC) (87%), TiVo series 1 (Linux 2.1.24-Ti
Vo-2.5) (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   0.53 ms 172.16.141.2
2   0.53 ms scanme.nmap.org (45.33.32.156)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/subm
it/ .
Nmap done: 1 IP address (1 host up) scanned in 30.44 seconds

```

**Figure 1.** nmap scan of scanme.nmap.org

## NMAP options

The journey of nmap and learning its options will be long and take some effort and will expand much further than this class. Let's start to explore the more specific options available within nmap. First, we are learning how to scan for TCP and UDP of specific ports. One issue I encountered while trying to scan certain ports was failing to include `-p <ports>` I instead attempted to clump the port numbers into the port scan I was attempting. So, in the case where I was scanning for TCP connect and wanted to scan ports 20-100 I tried `#nmap -sT20-100 scanme.nmap.org`. This was the first example of me not reading the man pages correctly.

```
(root@kali)-[~]
# nmap -sT -p 20-100,130-150,400-500 scanme.nmap.org
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-06 14:24 PDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.042s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 194 closed tcp ports (conn-refused)
PORT      STATE      SERVICE
22/tcp    open      ssh
25/tcp    filtered  smtp
80/tcp    open      http
135/tcp   filtered  msrpc
136/tcp   filtered  profile
137/tcp   filtered  netbios-ns
138/tcp   filtered  netbios-dgm
139/tcp   filtered  netbios-ssn
445/tcp   filtered  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 2.65 seconds
```

**Figure 2.** TCP connect scan of scanme.nmap.org

The next two individual actions I ran with nmap are detection of the operating system and IP protocol scan. Each of which are important. Figure 3 shows the result of the separate scans but in the same screenshot. In a video about using nmap created by Simplilearn the narrator explains and walks through about how to use nmap to find that port 445 is open. He explains that the eternal blue exploit might be a vulnerability that could be used and showed how that worked. This gave concrete evidence of how important nmap can be to scan our system as a Whitehat to ensure we fix vulnerabilities.

```
(root@kali)~# nmap -O scanme.nmap.org
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-06 20:59 PDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.043s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    filtered smtp
80/tcp    open  http
135/tcp    filtered msrpc
139/tcp    filtered netbios-ssn
445/tcp    filtered microsoft-ds
9929/tcp   open  nping-echo
31337/tcp  open  Elite
Aggressive OS guesses: Linux 3.2 (94%), Linux 4.4 (94%), DD-WRT v24-sp2 (Linux 2.4.37) (92%), Ac
tiontec MI424WR-GEN3I WAP (92%), Microsoft Windows XP SP3 (89%), Microsoft Windows XP SP3 or Win
dows 7 or Windows Server 2012 (89%), BlueArc Titan 2100 NAS device (86%), TiVo series 1 (Sony SV
R-2000 or Philips HDR112) (Linux 2.1.24-TiVo-2.5, PowerPC) (86%), TiVo series 1 (Linux 2.1.24-Ti
Vo-2.5) (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 16 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.34 seconds

(root@kali)~# nmap -sO scanme.nmap.org
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-06 20:59 PDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.015s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 252 filtered n/a protocols (proto-unreach)
PROTOCOL STATE SERVICE
1 open icmp
6 open tcp
17 open udp
47 open|filtered gre

Nmap done: 1 IP address (1 host up) scanned in 1.84 seconds
```

**Figure 3.** nmap OS detection scan on the top and IP protocol scan on the bottom.

## Packet capture

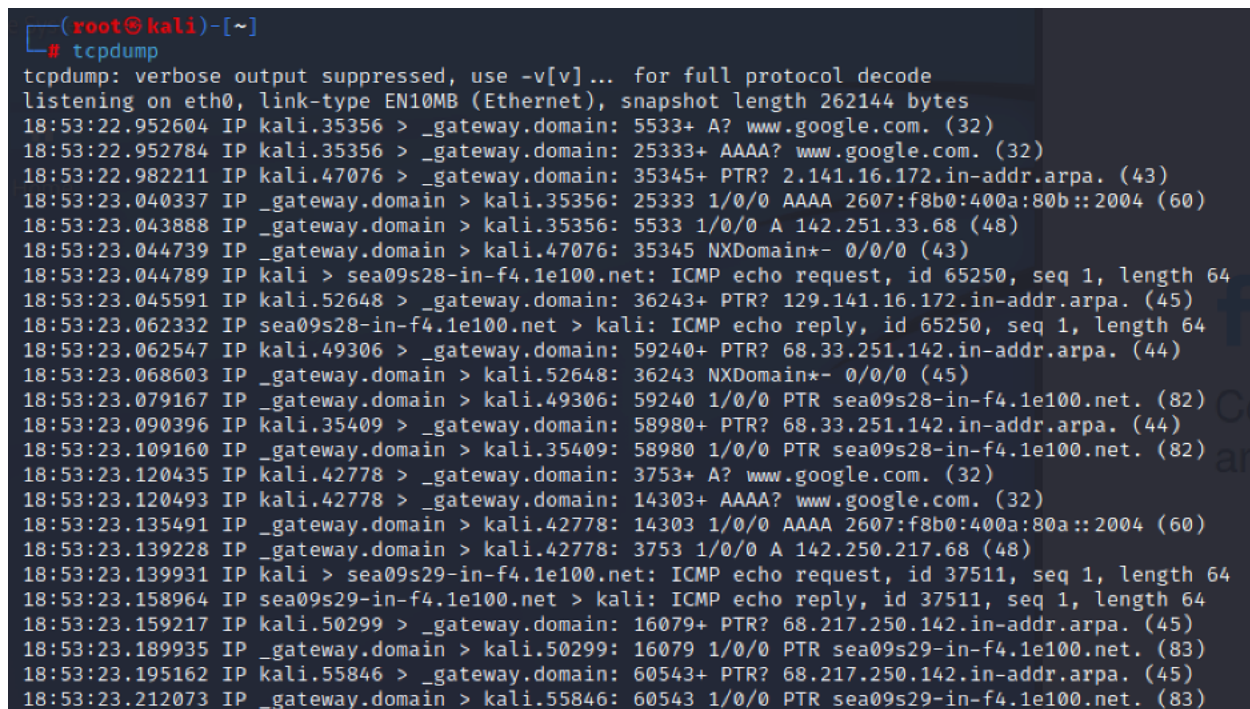
Capturing packets can be accomplished in multiple ways. This includes direct access to the channel of the network that includes ethernet cables, copper wire, and RF receiver. Having direct access can use different tools including a network tap, switches, and setting up a computer with dual ethernet ports to mirror a switch and capture packets as they travel through the machine. Another method is using a Wireless access point to capture packets with a WiFi card installed and set to promiscuous mode. The last method is having control of an endpoint This is the more practical case and the one we will begin investigating.

Our overall goal of capturing packets can range from checking the quality of service to checking for malicious activity that is happening on the network. The malicious actors can use it

for eavesdropping on data on the network all the way to espionage. This is a key place where having physical security is important to ensure no one has access to routers, network jacks, and endpoints. Just like many topics we have covered and will cover we can't tap or eavesdrop on any network that we do not own.

## tcpdump

Tcpdump is one of the command line tools that is a foundational tool used to capture packets. This can be configured in many ways to capture in and outbound packets or both at the same time. This can be done while writing to a file to later filter the results in ways to best fits our needs. We will be going over different commands explained in this module. The first command I entered to play with was 'tcpdump' this results in immediate results being displayed to the console. This produces results at such a rapid rate that it can be overwhelming to understand what is being displayed. Plus, we don't have a way to narrow down the current capture since the results aren't going to a file.



```
(root@kali)~#
# tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:53:22.952604 IP kali.35356 > _gateway.domain: 5533+ A? www.google.com. (32)
18:53:22.952784 IP kali.35356 > _gateway.domain: 25333+ AAAA? www.google.com. (32)
18:53:22.982211 IP kali.47076 > _gateway.domain: 35345+ PTR? 2.141.16.172.in-addr.arpa. (43)
18:53:23.040337 IP _gateway.domain > kali.35356: 25333 1/0/0 AAAA 2607:f8b0:400a:80b::2004 (60)
18:53:23.043888 IP _gateway.domain > kali.35356: 5533 1/0/0 A 142.251.33.68 (48)
18:53:23.044739 IP _gateway.domain > kali.47076: 35345 NXDomain*- 0/0/0 (43)
18:53:23.044789 IP kali > sea09s28-in-f4.1e100.net: ICMP echo request, id 65250, seq 1, length 64
18:53:23.045591 IP kali.52648 > _gateway.domain: 36243+ PTR? 129.141.16.172.in-addr.arpa. (45)
18:53:23.062332 IP sea09s28-in-f4.1e100.net > kali: ICMP echo reply, id 65250, seq 1, length 64
18:53:23.062547 IP kali.49306 > _gateway.domain: 59240+ PTR? 68.33.251.142.in-addr.arpa. (44)
18:53:23.068603 IP _gateway.domain > kali.52648: 36243 NXDomain*- 0/0/0 (45)
18:53:23.079167 IP _gateway.domain > kali.49306: 59240 1/0/0 PTR sea09s28-in-f4.1e100.net. (82)
18:53:23.090396 IP kali.35409 > _gateway.domain: 58980+ PTR? 68.33.251.142.in-addr.arpa. (44)
18:53:23.109160 IP _gateway.domain > kali.35409: 58980 1/0/0 PTR sea09s28-in-f4.1e100.net. (82)
18:53:23.120435 IP kali.42778 > _gateway.domain: 3753+ A? www.google.com. (32)
18:53:23.120493 IP kali.42778 > _gateway.domain: 14303+ AAAA? www.google.com. (32)
18:53:23.135491 IP _gateway.domain > kali.42778: 14303 1/0/0 AAAA 2607:f8b0:400a:80a::2004 (60)
18:53:23.139228 IP _gateway.domain > kali.42778: 3753 1/0/0 A 142.250.217.68 (48)
18:53:23.139931 IP kali > sea09s29-in-f4.1e100.net: ICMP echo request, id 37511, seq 1, length 64
18:53:23.158964 IP sea09s29-in-f4.1e100.net > kali: ICMP echo reply, id 37511, seq 1, length 64
18:53:23.159217 IP kali.50299 > _gateway.domain: 16079+ PTR? 68.217.250.142.in-addr.arpa. (45)
18:53:23.189935 IP _gateway.domain > kali.50299: 16079 1/0/0 PTR sea09s29-in-f4.1e100.net. (83)
18:53:23.195162 IP kali.55846 > _gateway.domain: 60543+ PTR? 68.217.250.142.in-addr.arpa. (45)
18:53:23.212073 IP _gateway.domain > kali.55846: 60543 1/0/0 PTR sea09s29-in-f4.1e100.net. (83)
```

Figure 1. result of entering just tcpdump with no flags or extra options or file being created.

The next command we will use will include a method to write the results to a file name of our choosing. This time I will be first\_capture.pcap. To do this we will include the flag -w after our tcpdump command. Figure two will have the results of the command line being entered. The image displays the number of packets captured as well as the packets received by the filter. If those first two numbers are different, then that means the difference in the value of packets that were received but not processed before exiting tcpdump. The last output will include the number of packets dropped by the kernel. In our instance it was zero.



```
(root@kali)-[~]
# tcpdump -w first_capture.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C2068 packets captured
2068 packets received by filter
0 packets dropped by kernel
```

Figure 2. command line directs the packets to a readable file.

## Viewing results

Viewing the results of the packets can be done by adding the flag for reading, followed by the file name, and then if we would like a verbose version of the results. Then ending with a pipe and the number of lines we would like to display for a small set of data from the file. Figures three and four will display the results we received from the first packets we captured. Figure three will be without verbose and figure four will include the verbose flag. As we can see while I was capturing packets, I went to Facebook. We can also see what method of communication we used and the size of each packet.

```
(root@kali)-[~]
# tcpdump -r first_capture.pcap | head -n20
reading from file first_capture.pcap, link-type EN10MB (Ethernet), snapshot length 262144
18:58:43.240447 IP kali.38967 > _gateway.domain: 52431+ A? www.google.com. (32)
18:58:43.240553 IP kali.38967 > _gateway.domain: 63014+ AAAA? www.google.com. (32)
18:58:43.369784 IP _gateway.domain > kali.38967: 52431 1/0/0 A 142.251.33.68 (48)
18:58:43.374732 IP _gateway.domain > kali.38967: 63014 1/0/0 AAAA 2607:f8b0:400a:806::2004 (60)
18:58:43.375365 IP kali > sea09s28-in-f4.1e100.net: ICMP echo request, id 57553, seq 1, length 64
18:58:43.392767 IP sea09s28-in-f4.1e100.net > kali: ICMP echo reply, id 57553, seq 1, length 64
18:58:43.393039 IP kali.37785 > _gateway.domain: 57402+ PTR? 68.33.251.142.in-addr.arpa. (44)
18:58:43.413137 IP _gateway.domain > kali.37785: 57402 1/0/0 PTR sea09s28-in-f4.1e100.net. (82)
18:58:45.346791 IP kali.46262 > _gateway.domain: 5348+ A? www.facebook.com. (34)
18:58:45.346921 IP kali.46262 > _gateway.domain: 65033+ AAAA? www.facebook.com. (34)
18:58:45.447784 IP kali.32881 > _gateway.domain: 19178+ A? www.facebook.com. (34)
18:58:45.464342 IP _gateway.domain > kali.46262: 5348 2/0/0 CNAME star-mini.c10r.facebook.com., A 157.240.3.35 (79)
18:58:45.467035 IP _gateway.domain > kali.32881: 19178 2/0/0 CNAME star-mini.c10r.facebook.com., A 157.240.3.35 (79)
18:58:45.467045 IP _gateway.domain > kali.46262: 65033 2/0/0 CNAME star-mini.c10r.facebook.com., AAAA 2a03:2880:f101:83:face:b00c:0:25de (91)
18:58:45.468062 IP kali.35604 > edge-star-mini-shv-01-sea1.facebook.com.https: Flags [S], seq 143311805, win 64240, options [mss 1460,sackOK,TS val 98526376 ecr 0,nop,wscale 7], length 0
18:58:45.469465 IP kali.35915 > edge-star-mini-shv-01-sea1.facebook.com.https: UDP, length 1357
18:58:45.470488 IP kali.35915 > edge-star-mini-shv-01-sea1.facebook.com.https: UDP, length 414
18:58:45.493128 IP edge-star-mini-shv-01-sea1.facebook.com.https > kali.35604: Flags [S.], seq 3009020269, ack 143311806, win 64240, options [mss 1460], length 0
18:58:45.493197 IP kali.35604 > edge-star-mini-shv-01-sea1.facebook.com.https: Flags [S], ack 1, win 64240, length 0
18:58:45.494998 IP edge-star-mini-shv-01-sea1.facebook.com.https > kali.35915: UDP, length 1232
```

Figure 3. without viewing 20 lines without the verbose flag

```
(root@kali)-[~]
# tcpdump -r first_capture.pcap -v | head -n20
reading from file first_capture.pcap, link-type EN10MB (Ethernet), snapshot length 262144
18:58:43.240447 IP (tos 0x0, ttl 64, id 9454, offset 0, flags [DF], proto UDP (17), length 60)
    kali.38967 > _gateway.domain: 52431+ A? www.google.com. (32)
18:58:43.240553 IP (tos 0x0, ttl 64, id 9455, offset 0, flags [DF], proto UDP (17), length 60)
    kali.38967 > _gateway.domain: 63014+ AAAA? www.google.com. (32)
18:58:43.369784 IP (tos 0x0, ttl 128, id 30384, offset 0, flags [none], proto UDP (17), length 76)
    _gateway.domain > kali.38967: 52431 1/0/0 www.google.com. A 142.251.33.68 (48)
18:58:43.374732 IP (tos 0x0, ttl 128, id 30385, offset 0, flags [none], proto UDP (17), length 88)
    _gateway.domain > kali.38967: 63014 1/0/0 www.google.com. AAAA 2607:f8b0:400a:806::2004 (60)
18:58:43.375365 IP (tos 0x0, ttl 64, id 11862, offset 0, flags [DF], proto ICMP (1), length 84)
    kali > sea09s28-in-f4.1e100.net: ICMP echo request, id 57553, seq 1, length 64
18:58:43.392767 IP (tos 0x0, ttl 128, id 30386, offset 0, flags [none], proto ICMP (1), length 84)
    sea09s28-in-f4.1e100.net > kali: ICMP echo reply, id 57553, seq 1, length 64
18:58:43.393039 IP (tos 0x0, ttl 64, id 54990, offset 0, flags [DF], proto UDP (17), length 72)
    kali.37785 > _gateway.domain: 57402+ PTR? 68.33.251.142.in-addr.arpa. (44)
18:58:43.413137 IP (tos 0x0, ttl 128, id 30387, offset 0, flags [none], proto UDP (17), length 110)
    _gateway.domain > kali.37785: 57402 1/0/0 68.33.251.142.in-addr.arpa. PTR sea09s28-in-f4.1e100.net. (82)
18:58:45.346791 IP (tos 0x0, ttl 64, id 36936, offset 0, flags [DF], proto UDP (17), length 62)
    kali.46262 > _gateway.domain: 5348+ A? www.facebook.com. (34)
18:58:45.346921 IP (tos 0x0, ttl 64, id 36937, offset 0, flags [DF], proto UDP (17), length 62)
    kali.46262 > _gateway.domain: 65033+ AAAA? www.facebook.com. (34)
```

Figure 4. Results with verbose mode.

The image will include the results of capture packets going out from the second packet capture and the results of packets coming in from the third capture. For each of these, I



included an even more verbose flag to show just how detailed we can get with these results. We can monitor whatever direction we desire or both directions at the same time. Figure five shows the results of in and out traffic independently.

Figure 5. In traffic on the left and out traffic on the right.

## Tcpdump filters

Next, we have the filters to narrow down the specific information we are looking for. We can filter to view UDP or TCP, what port we want, and the port range. The filter can even include Boolean searches to include AND, OR, and even NOT. This is where things become interesting and very specific and helps when we have a goal in mind. The following images will include examples of the power we have with filters. The first filter I picked shows data from only port 53 which is DNS. I also included a few new flags I haven't discussed which are -c5, -X, -K. The first one is to only show a count of 5 results. The X flag is used to print the hex and ascii dump and K flag not to validate the checksum.

Figure 6. The first example of filters. This shows DNS packet capture with hex and ascii dump.

Figure seven shows the example of filtering for just a port range. This will show the results of all results if they are included in that range. We can further narrow the search down by adding tcp/udp of a set of selected ports as well as a src and dst. This module shows a vast array of different options used to capture packets. Then after capturing the packets, we have the tools to filter in whatever manner we need. The skill to manipulate the results seems endless. The next step is learning to use the data in a way to help protect our network.

```

(root@kali)-[~]
# tcpdump -c10 -X -K -vvv -r third_capture.pcap portrange 20000-35000
reading from file third_capture.pcap, link-type EN10MB (Ethernet), snapshot length 262144
20:23:09.645665 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 1385)
kali.33308 > edge-star-mini-shv-01-sea1.facebook.com.https: UDP, length 1357
0x0000: 4500 0569 0000 4000 4011 5adf ac10 8d81 E..i..@.Z.....
0x0010: 9df0 0323 821c 01bb 0555 e00b cf00 0000 ...#....U.....
0x0020: 0108 2947 aa9f af8a 60c2 037a 0cbe 3c9e ..)G....`..z..<.
0x0030: 242e db11 df46 514f e2c0 7047 a59b 1b1d $....FQ0..pG....
0x0040: 9592 ebe2 a5ad 9b57 f008 1c36 0dab ac00 .....W...6....
0x0050: 0000 0076 41f9 7ab9 0c88 8c02 e124 1eb6 ...vA.z.....$..
0x0060: e574 c55d 3aa5 d798 2580 3342 3202 5d79 .t.]:...%.3B2.ly
0x0070: 7d3c 6267 c84d c546 7997 ec56 14be 16da }<bg.M.Fy..V....
0x0080: b0c2 84e7 1a33 16bc 99c8 10ee a0e5 8c48 .....3.....H
0x0090: 4276 6df9 8214 d123 76a2 8c61 a028 6d97 Bvm....#v..a.(m.
0x00a0: a5bd e69f bf5a 75ae 059e 0184 2554 9705 .....Zu.....%T..
0x00b0: cbcc e1c5 bc75 92ee 5c79 2191 bba9 b3eb .....u.. \y!.....

```

Figure 7. filter showing just results from port 20000-35000

## Sn1per

```

=====•x[2023-03-06](12:46)x•
CHECKING FOR WAF
=====•x[2023-03-06](12:46)x•

  W00f!

  404 Hack Not Found
  405 Not Allowed
  403 Forbidden
  502 Bad Gateway
  500 Internal Error

  ~ WAFW00F : v2.2.0 ~
  The Web Application Firewall Fingerprinting Toolkit

[*] Checking http://scanme.nmap.org
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7

```

## Example

## Sn1per

```
PORT  STATE SERVICE VERSION
80/tcp open  http      Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_vulners:
|_cpe:/a:apache:http_server:2.4.7:
|_CVE-2022-31813 7.5 https://vulners.com/cve/CVE-2022-31813
|_CVE-2022-23943 7.5 https://vulners.com/cve/CVE-2022-23943
|_CVE-2022-22720 7.5 https://vulners.com/cve/CVE-2022-22720
|_CVE-2021-44790 7.5 https://vulners.com/cve/CVE-2021-44790
|_CVE-2021-39275 7.5 https://vulners.com/cve/CVE-2021-39275
|_CVE-2021-26691 7.5 https://vulners.com/cve/CVE-2021-26691
|_CVE-2017-7679 7.5 https://vulners.com/cve/CVE-2017-7679
|_CVE-2017-3167 7.5 https://vulners.com/cve/CVE-2017-3167
|_CNVD-2022-73123 7.5 https://vulners.com/cnvd/CNVD-2022-73123
|_CNVD-2022-03225 7.5 https://vulners.com/cnvd/CNVD-2022-03225
|_CNVD-2021-102386 7.5 https://vulners.com/cnvd/CNVD-2021-102386
|_PACKETSTORM:127546 6.8 https://vulners.com/packetstorm/PACKETSTORM:127546 *EXPLOIT*
|_FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE8 6.8 https://vulners.com/githubexploit/FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE8 *EXPLOIT*
|_CVE-2021-40438 6.8 https://vulners.com/cve/CVE-2021-40438
|_CVE-2020-35452 6.8 https://vulners.com/cve/CVE-2020-35452
|_CVE-2018-1312 6.8 https://vulners.com/cve/CVE-2018-1312
|_CVE-2017-15715 6.8 https://vulners.com/cve/CVE-2017-15715
|_CVE-2016-5387 6.8 https://vulners.com/cve/CVE-2016-5387
|_CVE-2014-0226 6.8 https://vulners.com/cve/CVE-2014-0226
|_CNVD-2022-03224 6.8 https://vulners.com/cnvd/CNVD-2022-03224
|_8AFB43C5-ABD4-52AD-BB19-24D7884FF2A2 6.8 https://vulners.com/githubexploit/8AFB43C5-ABD4-52AD-BB19-24D7884FF2A2 *EXPLOIT*
|_4810E2D9-AC5F-5B08-BF83-DDAFA2F63332 6.8 https://vulners.com/githubexploit/4810E2D9-AC5F-5B08-BF83-DDAFA2F63332 *EXPLOIT*
|_4373C92A-2755-5538-9C91-0469C995AA9B 6.8 https://vulners.com/githubexploit/4373C92A-2755-5538-9C91-0469C995AA9B *EXPLOIT*
|_1337DAY-ID-22451 6.8 https://vulners.com/zdt/1337DAY-ID-22451 *EXPLOIT*
|_0095E929-7573-5E4A-A7FA-F6598A35E8DE 6.8 https://vulners.com/githubexploit/0095E929-7573-5E4A-A7FA-F6598A35E8DE *EXPLOIT*
```

## Example

```
===== *x[2023-03-06](12:49)x*
RUNNING SCOPE NETWORK VULNERABILITY SCAN
===== *x[2023-03-06](12:49)x*
P4 - LOW, SSH Version Disclosure, scanme.nmap.org, [+] 45.33.32.156:22 - SSH server version: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2015-5600 8.5 https://vulners.com/cve/CVE-2015-5600
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2015-6564 6.9 https://vulners.com/cve/CVE-2015-6564
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2018-15919 5.0 https://vulners.com/cve/CVE-2018-15919
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2021-41617 4.4 https://vulners.com/cve/CVE-2021-41617
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2020-14145 4.3 https://vulners.com/cve/CVE-2020-14145
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2015-5352 4.3 https://vulners.com/cve/CVE-2015-5352
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2015-6563 1.9 https://vulners.com/cve/CVE-2015-6563
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2022-31813 7.5 https://vulners.com/cve/CVE-2022-31813
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2022-23943 7.5 https://vulners.com/cve/CVE-2022-23943
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2022-22720 7.5 https://vulners.com/cve/CVE-2022-22720
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2021-44790 7.5 https://vulners.com/cve/CVE-2021-44790
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2021-39275 7.5 https://vulners.com/cve/CVE-2021-39275
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2021-26691 7.5 https://vulners.com/cve/CVE-2021-26691
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2017-7679 7.5 https://vulners.com/cve/CVE-2017-7679
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2017-3167 7.5 https://vulners.com/cve/CVE-2017-3167
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CNVD-2022-73123 7.5 https://vulners.com/cnvd/CNVD-2022-73123
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CNVD-2022-03225 7.5 https://vulners.com/cnvd/CNVD-2022-03225
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CNVD-2021-102386 7.5 https://vulners.com/cnvd/CNVD-2021-102386

=====
*(^°... Sc0pe Vulnerability Report by @xer0dayz _..°°))$*
=====
Critical: 0
High: 1
Medium: 95
Low: 1
Info: 3
Score: 294
=====
P2 - HIGH, Clear-Text Protocol - HTTP, http://scanme.nmap.org:80/, HTTP/1.1 200 OK
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2015-5600 8.5 https://vulners.com/cve/CVE-2015-5600
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2015-6564 6.9 https://vulners.com/cve/CVE-2015-6564
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2018-15919 5.0 https://vulners.com/cve/CVE-2018-15919
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2021-41617 4.4 https://vulners.com/cve/CVE-2021-41617
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2020-14145 4.3 https://vulners.com/cve/CVE-2020-14145
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2015-5352 4.3 https://vulners.com/cve/CVE-2015-5352
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2015-6563 1.9 https://vulners.com/cve/CVE-2015-6563
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2022-31813 7.5 https://vulners.com/cve/CVE-2022-31813
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2022-23943 7.5 https://vulners.com/cve/CVE-2022-23943
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2022-22720 7.5 https://vulners.com/cve/CVE-2022-22720
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2021-44790 7.5 https://vulners.com/cve/CVE-2021-44790
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2021-39275 7.5 https://vulners.com/cve/CVE-2021-39275
P3 - MEDIUM, Components with Known Vulnerabilities - NMap, scanme.nmap.org, CVE-2021-26691 7.5 https://vulners.com/cve/CVE-2021-26691
```

## Example

### Recommendations

Creating a secure website involves a couple of key steps. First, choose a host that is reliable and provides reliable support. Make sure the host has up-to-date security protocols and services. Second, use secure coding techniques and tools. Choose an encryption scheme for your website, and make sure your passwords are not easily guessable. Third, regularly monitor

the website for suspicious activity and update your website's security. Finally, use a service such as Google's reCAPTCHA to protect the website from malicious bots. By following these steps, you can ensure your website is secure and protected from malicious attacks.