A Study on Portable Common Vulnerability Scanner on Raspberry Pi

Akash Joshi, Dr. Shivprasad Patil, Mitali Suryawanshi, Ashwini Doke, Ankush Sharma
Information technology,
Savitribai Phule Pune University

***Abstract*— Nowadays, finding vulnerabilities in web applications can not only be relied on manual effort. Automation plays a crucial role in vulnerability discovery and estimating application's internal structure or work flow based on responses revived by large amount of automated requests made at a faster pace or maybe slower pace depending on the security researcher's/penetration tester's decision about the way to interact with the application, keeping in mind not to trigger any already present security solutions like IDS, Firewall etc. Setting up runtime environment for tools needed to run the tools (scripts) needed for finding vulnerabilities is a tiresome task. Setting up vulnerability scanner on a credit- card sized computer like raspberry pi and availing the functionalities served is easier for the security researcher/penetration tester/bug bounty hunter. Security researcher/Penetration tester love using terminal because it's more flexible to adapt to changes rather than GUI but it's also a relief to get common and small tasks done by a just a click or two. Wasting time on non-productive, in sense of time consuming and still traditional techniques being used, bugs findings is not efficient for security researcher as finding new path or bypassing new patches needs much more focus and time. Swarm like technology (FaaS) in a single software technology is made available by big companies but costs a lot. This system presents a hardware such that we can just attach it to the main client device and run the application, if swarm of hardware is used then serverless application, solution to web application vulnerability researching at comparatively lower costs and that too Do It Yourself customizable (open source).**

Keywords*— *Web app, vulnerability, scanner, raspberry pi, OWASP, CVE.

I. INTRODUCTION

Task of scanning vulnerabilities dependent on automated scanners is not even the half job done but it still is part of the process to focus on finding new attack vectors. Chaining of bugs which mostly leads to a critical impact (potential, if employed further) on the target's asset. Small findings like open redirect etc... Combined with other small impact vulnerabilities helps in forming a better and more impactful attack vector. Web Application Vulnerability Scanners when used without the limitation of where (on whose device) to use and the resulting data reports to manage becomes a hassle-free process to do. Such Scanners which has its own file database system (for e.g., JSON based), serves interactive GUI web application over ethernet2usb bridge on the client device.

Making use of the compact credit-card sized computer's processing capabilities and external storage this system is a pocket-friendly web application vulnerability scanner.

New attack vectors found by black-box approach needs automation plus manual assessment. A lot of time vulnerability scanner gives false alerts this can be verified through manual assessment. Before report submission, manual assessment is needed and the exact report of automated scanners is not relayed directly.

A lot more can be done using multiple number of single board computers i.e. building a more powerful scanner with a lot more varieties of vulnerabilities added and more clean automation as processing is also distributed in a such rigs [1][2].

A. INTRODUCTION to PENETRATION TESTING

A penetration testing also called as pentesting is a practice of testing computer system, network or web application to check for exploitable vulnerabilities. It is also used to find security vulnerabilities. Penetration testing can be manually performed or can be automated with software applications. Security testing that uncovers vulnerabilities (bugs), threats, risks in software application, web application or network application. The process involves Information gathering, Scanning, Discover vulnerabilities, Exploitation and Report generation.

B. INTRODUCTION to VULNERABILITY SCANNER

A vulnerability scanner is a computer program designed for assessing computers, networks or applications for known flaws. They are used for finding and detecting vulnerabilities arising from mis-configurations or flawed programming within a network-based asset such as a firewall, router, web server, application server, etc. Vulnerability scanners allow us authenticated and unauthenticated scans. Vulnerability scanners are generally available as SaaS (Software as a service); provided through internet and delivered in the form of web application. The vulnerability scanner has the ability to personalize vulnerability reports as well as the installation of software, open ports, certificates and other host information that can be objected as part of its workflow.

C. COMMON WEB APPLICATION SECURITY MISCONFIGURATIONS and FLAWS

Some of the OWASP TOP 10 [3] and other vulnerabilities, which have critical impact too, are mentioned bellow:

1) Cross Origin Resource Sharing (CORS):

CORS happens to be a W3 specification that permits cross domain communications from the browser. It function by adding new HTTP Headers which describe the origins that are allowed cross domain information sharing. In other words, CORS is employed to relax the 'Same Origin Policy' for legitimate and trusted requests. It's an important feature of Web 2.0 to support APIs that are exposed via web services to be accessible.

However, it also provides potential for cross-domain attacks, if a website's Cross Origin Resource Sharing policy is poorly configured and implemented.

2) Potentially Dangerous HTTP Methods:

There are a variety of official (standards compliant) HTTP methods:
OPTIONS, HEAD, GET, POST, PUT, DELETE, TRACE, CONNECT

To retrieve static and dynamic data content a web server supports the HEAD, GET and POST methods (which enable WebDAV on a web server will add support for the PUT and DELETE methods).
TRACE and TRACK are methods which may be used for debugging purposes. It repeats the content of a request, and an attacker could steal credentials by employing a client-side attack.
These HTTP methods shouldn't be supported on public web servers, as they increase the attack surface.

3) Subdomain Takeover:

A subdomain takeover is taken into account as a high severity threat and boils right down to the registration of a domain by somebody else (with bad intentions) By doing this, the full control of subdomain can be taken by hacker. By using external services Subdomain Takeover might be done like Desk, Squarespace, Shopify, GitHub, Tumblr, and Heroku.

4) Clear Text Password Submission:

The software transmits sensitive or security-critical data in cleartext during a channel which will be sniffed by unauthorized actors.

5) Critical File:

A web security vulnerability that permits an attacker to read arbitrary files on the server that's running an application. This might include application code and data, credentials for back-end systems, and sensitive OS files

6) Open Redirect:

Web applications return a 301 or 302 response code to instruct web browsers to redirect to another URL. An open redirect vulnerability can exist when a web application leverages unsensitized user-supplied data (intended or not) to determine the destination of the redirection.

A vulnerable application allows an attacker to craft a link having a destination URL that causes users to be redirected to the attacker's choice of sites. The link would appear to be benign to most people and when clicked, the redirection occurs seamlessly so users likely won't even notice it happened.

7) SSL Testing:

Checks a server's service on any port for the support of TLS/SSL ciphers, protocols also recent cryptographic flaws

8) SPF and DMARC:

An SPF record happens to be a Domain name Service (DNS) record that identifies which mail servers are permitted to send email on behalf of your domain. the aim of an SPF record is to stop spammers from sending messages with sender addresses of your domain. Missing SPF record allows hackers to send spam emails by using an email address that has your domain name as its suffix.

DMARC (Domain-based Message Authentication, Reporting and Conformance) is an email authentication protocol that's designed to protect against impersonation or spoofing.

The technologies build upon Sender Policy Framework (SPF) and Domain Keys Identified Mail (DKIM).

II. RELATED WORK

Vulnerability Scanner Pi:

In this project, you will be scanning a network using OpenVAS to look for cyber vulnerabilities. In addition, you will examine how they are displayed in OpenVAS; reviewing in more detail specific vulnerabilities discovered on your machine. You can also conduct further research on each of the vulnerabilities you find to discover how they originated.

Kali Linux OS used in this project.

This focusses on network scanning but we focused on web application vulnerability scanning with some features really useful for the user like newly published CVE detection and notification, report submission with PoC etc...

III. PROPOSED SYSTEM

This web application scanner, which is setup on a raspberry pi and used over the ethernet2usb bridge by the client(browser or electron app), helps the user(security researcher/penetration tester/bug bounty hunter) to get interactive options to detect common vulnerabilities like CORS, Host Header Injection, Potentially Dangerous HTTP methods, Subdomain Takeover, Clear Text Password Submission, Critical File Check, Open Redirection Testing, SPF and DMARC additionally integration of exploit suggestor by metasploit based on port scan results by NMAP and if user wants more categories, ones those require more processing power, of vulnerability scanning to cover then a raspberry pi cluster/rig/swarm is much more powerful to use.

We are using two single board computers (Raspberry Pi in our case) to demonstrate above mentioned basic vulnerability detection. One for rendering UI and the other for handling, executing scripts and tools.

Rather than using hundreds of tools from Linux distribution like Kali OS [4] we are using DietPi OS which is very lightweight and only necessary tools and configuration can be present.

Newly published web application related CVE is notified to the user via preferred channels.

Other than exporting vulnerability report [5], auto submission is also a part of this web application. Currently supported platforms are Hackerone and Bugcrowd for report generation and submission.

1) Report Generation

Shell script that does this report generation and submission takes markdown file as input and according to pre-defined syntax the new parsed output data is sent via live session (Reporting account's username and password involved).2FA authentication is supported.

NO DUMMY DEMONSTRATION WERE MADE USING THE ABOVE SHELL SCRIPT.

AS IT MAY LEAD TO FALSE REPORT SUBMISSION AND THEN LEAD TO THE ACCOUNT GETTING BANNED.

2) Graphical Interface Overview

1. Input for domain name and then displaying all recon (information gathering) data.
2. Scanner section which includes above mentioned checks
   CORS, Host Header Injection,
   Potentially Dangerous HTTP
   methods, Subdomain Takeover,
   Clear Text Password
   Submission, Critical File
   Check,
   Open
   Redirect,
   SSL
   Testing,
   SPF and DMARC
3. Report generation and auto submission based on H1 (hackerone) or BC (bugcrowd) program directory.
4. Target and Vulnerability Checklist
   i. Info tab – Recon data
   ii. Checklist - Discovered Bugs

If serverless was to be opted then docker implementation is the best currently known way to establish OpenFaaS framework.[2] Template rendering engine (ejs) allows using logical statements in html and this helped a lot in balancing the flow of requests and responses. Nodejs is non-blocking, asynchronous then too the single board computer (raspberry pi) has limitations.

JSON data exchange is a good way to keep up with the data generated and used in backend as well as frontend instead of a heavy database management system this JSON database can be replicated and distributed or mirrored over backup servers.

Frontend parses and utilizes this JSON received from server, for showing current available data, every time the template is rendered. XML Http Request is utilized to retrieve such data instead of template rendering because every time render function is called, large JavaScript object needs to be passed and maintaining database this way is not efficient.

Updating JSON needs to read JSON file, update locally parsed JSON data, write (not append but erase and write) the whole JSON data to the same file it was read from before.

A. Hardware Configuration:

1) Using NFQUEUE and libnetfilter_queue

NFQUEUE is an iptables and ip6tables target which represents the decision on packets to userspace software. For example, the following rule will ask for a decision to a listening userspace program for all packet going to the box:

```
iptables -A INPUT -j NFQUEUE --queue-num 0
```

To get the messages from kernel in userspace, software must use libnetfilter_queue to connect to queue 0 (the default one). Then it must issue a verdict on the packet.

2) Inner working

For understanding NFQUEUE, the easiest way is to understand the working inside Linux kernel. The packet after reaching an NFQUEUE target it is then en-queued to the queue corresponding to the number given by the --queue-num option. The packet queue is an executed as a chained list with element being the packet and metadata (a Linux kernel skb):

- It is in the form of fixed length queue which is implemented as a linked-list of packets
- Packet Storing is done which are indexed by an integer
- When userspace issue a verdict to the corresponding index integer a packet is released
- No packet can be enqueued to it, when queue is

full This has some implication on user space side:

- Userspace can read multiple packets and wait to give a decision. If the queue is not full there is no change in behaviour.
- Packets can be convicted in any order. Userspace can read packet 1,2,3,4 and convicted at 4,2,3,1 in that order.
- Slow conviction will result in a full queue. Instead of en-queuing them, kernel will then drop incoming packets

B. TECHNOLOGY USED

Raspberry Pi hosts the server for vulnerability scanner application.

NodeJS used in building server program and some Python scripts are run in node’s main process as child process. Results from these scripts returned to Node server and results crafted and rendered using node’s template engine EJS as middleware.

ExpressJS used with NodeJS to create easy application object from express constructor function.

```
var Express =
require(‘express’); var app =
Express ();
```

Packet manipulation in python scripts requires a good network library. Scapy is the library best suitable and available at this point of time.

Such packet manipulation needs the packets to queue up. Then performs operations requested and finally forward the packet to the original desired address. IP tables in raspberry pi are configures for the same. Such that packet from device passes through raspberry pi to Internet followed by local router.

- Summary of technologies required:
    - Raspberry Pi (Hardware)
    - NodeJS + Python (Server Process)
    - Golang
    - ExpressJS (HTTP based framework)
    - Template Engine (Rendering Middleware)

- Scapy (Packet Manipulation and Analysis library)
- Other open source scripts for Reconnaissance.
  - Assetfinder
  - Masscan / Namp
  - Subtake
  - Virtual Host Discovery
  - DNSScan
  - Gitrob
  - S3 bucket finder
  - Webscreenshot

C. Experiments

Checking false positives using mathematical approach [6][7][8] along with visually analysing by using webscreenshot to get better analysis results.

IV. CONCLUSION

This system allows the user (security researcher/penetration tester/bug bounty hunter) to fetch easy bounties as well as formulate new attack vectors by manual assessment based on the automated responses. This involves fewer accessories to work with the single-board computer and mirroring of single-board computer desktop environment GUI is avoided by setting up node server to server requests in a non-blocking asynchronous manner.

This will help bug bounty hunters, security researchers and penetration testers to do security testing and vulnerability assessment on the go without any configuring issues and network hassle.

REFERENCES

[1] Cheng Wang, Xin Liu, Xiaokang Zhou, Rui Zhou, Dong
[2] Qingquan, Mingsong Wang4 and Qingguo Zhou FalconEye: A High-performance Distributed Security Scanning System (2019) https://blog.alexellis.io/your-serverless-raspberry-pi-cluster/
[3] VULNERABILITY SCANNER PI https://www.cyberpiprojects.com/vulnerability-scanner-pi/
[4] Parthajit Dholey and Anup Kumar Shaw: "OnlineKALI: Online Vulnerability Scanner" (2018)
[5] Shailendra Singh and Karan Singh: Performance Analysis of Vulnerability. Detection Scanners for Web Systems (2018)
[6] Balume Mburano, Weisheng Si: Evaluation of Web Vulnerability Scanners Based on OWASP Benchmark (2018)
[7] Prof. Smita Patil, Prof. Nilesh Marathe and Prof. Puja Padiya: Design of Efficient Web Vulnerability Scanner (2016)