

Autonomous Penetration Testing for Web Applications Using Raspberry Pi

Rakesh R¹, Jeeva N²

Electronics and Communication Engineering, Dr.NGP Institute of Technology, Coimbatore, India

Abstract:

During twenty first century, automation business is the the game for those who involves. It applies to more general areas of business such as manufacturing and control but it also applies to more technical areas of IT such as web application security. Always business is not fully automated, it needs more duration, work, and cost – requirements that cannot be squandered. While accomplishing business tasks like sound labour in web application, security and penetration testing, it awakes a considerable burden. Here introducing a highly customizable USB attack platform, using Raspberry Pi Zero or Raspberry Pi Zero W with cost efficiency. To identify common web application security vulnerabilities and to determine their presence in web applications. The need of this project is to implement an autonomous hacking tool for web application testing. The following chapter deals with the detailed survey of the existing technologies

Keywords — Raspberry Pi Zero, HID covert channel Frontdoor/Backdoor.

I. INTRODUCTION

A. Introduction White HAT hacking and its importance

White hat explains a hacker (or, if you like, cracker) who identifies a security weakness during a computing system or network however, instead of taking malicious advantage of it, exposes the weakness during a approach that may permit the system's homeowners to mend the breach before it may be taken by others (other types of hackers.) ways of telling the homeowners regarding it vary from an easy telephone call through sending associate e-mail note to a Webmaster or administrator all the way to departure associate electronic "calling card" within the system that creates it obvious that security has been broken.

While white hat hacking may be a hobby for a few, others as a business. Therefore, a white hat hacker may work as a consultant or be a permanent worker on a company's payroll. A decent several white hat hackers are past black hat hackers. The name derived from old Western movies, wherever hero's typically wore white hats and therefore the "bad guys" wore black hats. White hat hacking involves a good deal of problem resolution, in addition as communication skills. A white hat hacker additionally needs a balance of intelligence and common sense, sturdy technical

and structure skills, impeccable judgment and therefore the ability to stay cool under pressure.

At identical time, a white hat has to assume sort of a black hat hacker, with all their wicked goals and devious skills and behavior. Some top-rate white hat hackers are former black hat hackers who got caught, and for numerous reasons determined to go away a lifetime of crime behind and place their skills to figure in a positive (and legal) approach.

There aren't any normal education criteria for a white hat hacker— each organization will impose its own needs on it position— however a bachelor's or degree in information security, computer science or maybe mathematics provides a powerful foundation.

B. Need For The Project

During twenty first century, automation business is the the game for those who involves. It applies to more general areas of business such as manufacturing and control but it also applies to more technical areas of IT such as web application security. Always business is not fully automated, it needs more duration, work, and cost – requirements that cannot be squandered.

While accomplishing business tasks like sound labour in web application, security and penetration testing, it awakes a

considerable burden.. Gazing internet vulnerability testing, several resources are needed during:

- Project scoping
- Gathering Information
- Scanning for internet application vulnerabilities
- Vulnerability identification and validation
- Reporting
- Remediation

In any given organization these factors usually involve various people: Developers, QA analysts, project managers, network directors, internet application developers, data security managements, auditors, and management. Even third-party vendors are usually pulled into internet security assessment projects. With this several highly-paid staff members operating toward a standard goal, each business should change the maximum amount as doable to avoid confusion and costly bills.

The question becomes: Why? Why is automation important? Each state of affairs is completely different however there are some commonalities. For starters, you run the risk of duplicated efforts once redundant tests are performed. Once you have various complicated internet applications as many of today's on-line businesses do, this may add up to a substantial quantity of unneeded work. Another issue that management does not totally perceive is that there is not enough information or time to perform manual internet vulnerability testing on all internet applications all the time. Nobody is that sensible a lot of less that sensible at time and project management. If internet application security checking isn't machine-controlled employing a well-trying machine-controlled internet application security scanner that will test for thousands of potential security flaws, suppose if not all of the intense internet application vulnerabilities is unnoted. Internet security testing goes from being a apparently benign IT project to a significant business liability.

For example, imagine a custom created internet primarily based enterprise resource coming up with (ERP) system. Such system would have a whole bunch, if not thousands of visible entry points or attack surfaces and plenty of different "under the hood" that require to be checked for internet application vulnerabilities like SQL injection and cross-site scripting.

Using real world numbers, imagine the ERP system has two hundred entry points that require to be checked against one hundred completely different internet application vulnerability

variants meaning that the penetration tester must launch a minimum of twenty thousand security tests. If each check had to take simply five minutes to finish, it might take {a web/an internet/an online} security specialist around 208 business days to finish a correct web application security audit of associate ERP system.

An automated internet application security scanner like Net sparker will scan a far larger custom ERP systems against a far larger variety of internet application vulnerability variants in a very matter of hours and in contrast to a person's, an automatic security scanner won't forget to scan associate input parameter or get bored whereas attempting completely different variations of a specific attack.

When doing a manual internet application security check, you're additionally limiting the penetration to variety of best-known vulnerabilities best-known to the penetration tester. On the opposite hand, once using an automatic internet vulnerability scanner like Net sparker {you are/you're} ensuring that each one parameters are being checked against all kind of internet application security variants. By using Net sparker {you are/you're} additionally making certain that no false positives are reportable within the internet application security scan results, thus you do not have to be compelled to allocate time to validating detected vulnerabilities.

Underscoring the importance of vulnerability testing automation are the popular information security studies. Year when year this analysis points to an equivalent underlying causes of data risks like scarce resources, improper vision, and unenlightened management. Every of those components are addressed by automating security testing processes.

There's no excellent way to check for internet security vulnerabilities. However, one issue is for sure: going concerning it manually associated hoping on employees experience alone is an exercise in uselessness that you simply cannot afford to require on because it {might/it'd/it would} price your business lots of cash and a few internet application vulnerabilities might go undetected . Do the best for the business and integrate automation into {the internet/the online/the net} vulnerability testing discussion and into web applications software package development life cycle. Once using an automatic internet application security scanner you discover additional and higher vulnerabilities.

There are problems wherever automation won't assist you and manual testing must happen, however you do not wish your security team to see associate input for one hundred completely

different potential problems one HTTP request at a stretch or by attempting to analyse the output of a fuzzer. Free your team members' time so that they will focus their work to the tasks that really can have the benefit of their experience.

C. Problem Identifications

Attackers have an over-growing list of vulnerabilities to take maliciously gain access to your net applications, networks and servers. whether or not you're a novice Word Press user or a complicated hosting service, if determined then an attacker can notice any vulnerability you've didn't patch and use it for their advantage.

By users, by security researchers, by attackers new vulnerabilities are discovered all the time; Every time changes are created at any level of the infrastructure, there's the potential for new vulnerabilities to be created.

The following is a list of best-known net application and network-layer vulnerabilities which will be automatically detected by this project. While most of the illustrated examples can discuss PHP coding as a result of its overwhelming quality on the net, the ideas additionally apply to any artificial language. The attacks explained during this project are:

- Remote code execution
- SQL injection
- Format string vulnerabilities
- Cross site Scripting (XSS)
- Username enumeration

Considering the poor programming approach that results in these attacks, the article provides some real samples of common products that had these same vulnerabilities in the past. Some countermeasures are offered with every example to assist prevent future vulnerabilities and consequent attacks.

D. Objective

- Recognize common internet application security vulnerabilities and the way to see if they're present in internet applications.
- Recognize internet application design assumptions and the way to use them
- Be awake of the capabilities of various Browser Proxies
- Be awake of the capabilities of various Penetration Testing tools
- Be ready to discover Access control Vulnerabilities
- Be ready to discover SQL Injection Vulnerabilities
- Be ready to discover Cross-Site Scripting (XSS) Vulnerabilities

- Be ready to discover Authentication and Session Vulnerabilities
- Be ready to check internet application security

E. Organization Of The Project

The report is divided into five chapters, every depicts numerous aspects of the project. This project may be an extremely customizable USB attack platform, based on an occasional price Raspberry Pi Zero or Raspberry Pi Zero W.

This project features:

- HID covert channel Frontdoor/Backdoor | Get remote shell access to Windows targets via HID devices)
- Windows 10 Lock picker | Unlock Windows boxes with weak passwords (fully automated)
- Stealing Browser Credentials | Dumps keep Browser Credentials and copy them to the built-in flash drive
- WiFi Hotspot | SSH access (Pi Zero W only), supports hidden ESSID
- Client Mode | Relays USB net attacks over WiFi with internet access (MitM)
- USB device | Works with Windows Plug and Play support
- Device Types:
- HID covert channel communication device | Frontdoor/Backdoor
- HID Keyboard/Mouse
- USB Mass storage | presently solely in demo setup with 128 MB drive
- RNDIS | Windows Networking
- CDC ECM | MacOS / Linux Networking
- Bash primarily based payload scripts | See payloads/ subfolder for example Responder
- John the ripper jumbo | Pre-compiled version ready to go
- AutoSSH integration | for simple reverse ssh tunnels
- Auto attack | this project automatically boots to standard shell if associate OTG adapter is connected
- LED state feedback | with a simple bash command (led_blink)

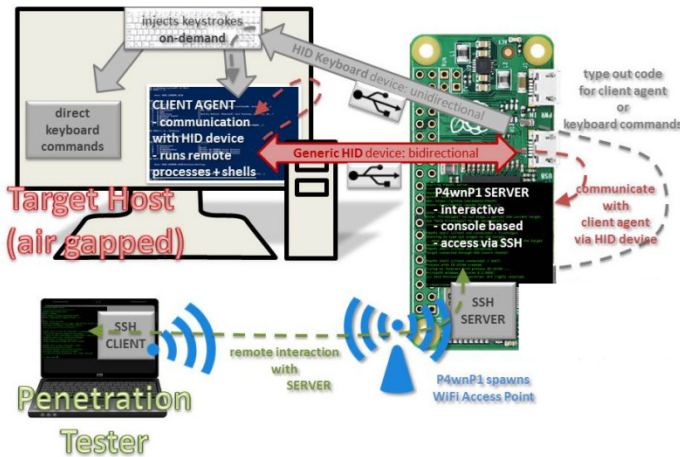


Fig. 1. Automated backdoor entry

II. EXISTING SYSTEM

What is Vulnerability”?

Vulnerability is a weakness in the web application. The cause of such a “weakness” can be bugs in the application, an injection (SQL/ script code) or the presence of viruses.

What is “URL manipulation”?

Some web applications communicate additional information between server and the client (browser) in the URL. Changing little information in the URL may sometimes lead to unintended behaviour by the server.

What is “SQL injection”?

SQL injection is the process of inserting SQL statements through the web application user interface into some query that is then executed by the server.

What is “XSS (Cross Site Scripting)”?

When a user inserts HTML/ client-side script in the user interface of a web application and this insertion is visible to other users, it is called XSS.

What is “Spoofing”?

The creation of hoax look-alike websites or emails is called spoofing.

A. Security testing approach:

In order to perform a useful security test of a web application, the security tester should have good knowledge of the HTTP protocol. It is important to have an understanding of how the client (browser) and the server communicate using HTTP. In addition, the tester should at least know the basics of SQL injection and XSS. Hopefully, the number of security defects present in the web application may not be high. However, being able to accurately describe the security defects with all the required details to all concerned can definitely help.

B. Password cracking:

The security testing on an internet application are often set out by “password cracking”, so as to log in to the non-public are as of the application, one will either guess a username/ password or use some password cracker tool for identical. Lists of usernames and passwords are available at the side of open source password crackers. If the online application doesn't enforce a posh password (e.g. with alphabets, number and special characters, with a minimum of a needed range of characters), it's going to not take terribly long to crack the username and password. If username or password is hold on in cookies while not encrypting, attacker will use completely different strategies to steal the cookies then info stored within the cookies like username and password. For a lot of details see article on “Website cookie testing”.

C. URL manipulation through HTTP GET methods:

The tester ought to check if the application passes necessary information within the query string. This happens once the application uses the HTTP GET technique to pass information between the client and also the server. The information is passed in parameters within the query string. The tester will modify a parameter value within the query string to ascertain if the server accepts it via HTTP GET request user data is passed to server for taking information or authentication. Attacker will manipulate each input variable passed from this GET request to server so as to urge the specified data or to corrupt the information. In such conditions any uncommon behaviour by application or internet server is that the doorway for the attacker to urge into the application.

D. SQL Injection:

The next issue that ought to be checked is SQL injection entering a single quote (‘) in any textbox ought to be rejected by the application. In case, if the tester encounters a database error, it implies that the input of user is inserted in some query that is then executed by the application. Suppose, the application is prone to SQL injection. SQL injection attacks are very essential as attacker will get very important information from server information to ascertain SQL injection entry points into your internet application, find out code from your code base wherever direct MySQL queries are executed on database by accepting some user inputs. If user input data is crafted in SQL queries to query the database, attacker will inject SQL statements or a part of SQL statements as inputs of user to extract very important data from database though attacker is fortunate to

crash the applying, from the SQL query error shown on browser, offender will get the information they're searching for. Special characters from user inputs ought to be handled/escaped properly in such cases.

E. Cross web site Scripting (XSS):

The tester ought to to boot check the net application for XSS (Cross web site scripting). Any HTML e.g. <HTML> or any script e.g. ought to not be accepted by the application. If it is, the application may be vulnerable to associate attack by Cross web site Scripting. Attacker will use this technique is to execute malicious script or URL on victim's browser mistreatment cross-site scripting, attacker will use scripts like JavaScript to steal user cookies and information keep within the cookies. Many internet applications get some user information and pass this information in some variables from completely different pages. E.g.: <http://www.examplesite.com/index.php?123&query=xyz> Attacker will simply pass some malicious input or as a '&query' parameter which may explore necessary user/server information on browser Important: throughout security testing, the tester ought to be terribly careful to not modify any of the following: Configuration of the application or the server Services running on the server Existing user or client information hosted by the application Additionally, a security check ought to be avoided on a production system.

The purpose of the security check is to find the vulnerabilities of the net application so the developers will then take away these vulnerabilities from the application and create the net application and data safe from unauthorized actions.

- Share72
- Tweet
- +110
- Pin
- Share24

F. Internet application security testing with Veracode

Veracode provides cloud security applications and services that secure the applications driving today's businesses. Providing a robust combination of automation, process and speed, Veracode seamlessly integrates testing into software development, serving to to eliminate vulnerabilities additional simply and cost-effectively.

Veracode simplifies internet application security testing with a cloud-based resolution that needs no investment in hardware, software or security experience.

Developers will access code review tools on-demand and scale effortlessly to fulfill secure internet application development deadlines.

Developers will access Veracode's internet application security testing tools through an internet portal. check results are came quickly and prioritized in a very Fix-First Analysis that identifies each the foremost urgent flaws and also the ones that may be fixed most quickly, permitting developers to optimize efforts and save extra resources for the enterprise. FEATURES OTHER TOOLS OUR TOOL RNDIS, CDC ECM, HID , serial and Mass storage support supported, usable in many mixtures, Windows class driver support (Plug and Play) in most modes supported, usable in most mixtures, Windows class driver support (Plug and Play) altogether modes as composite device Target to device communication on covert HID channel no Raw HID device permits communication with Windows Targets (PowerShell 2.0+ present) via raw HID There's a full automatic payload, permitting to access Rakesh Hack Portal bash via a custom PowerShell console from target device (see 'hid_frontdoor.txt' payload). An additional payload supported this method, permits to show a backdoor session to Rakesh Hack Portal via HID covert channel and relaying it via WiFi/Bluetooth to any SSH capable device (bridging airgaps, payload 'hid_backdoor.txt') Mouse emulation no Supported: relative Mouse positioning (most OS, as well as Android) + ABSOLUTE mouse positioning (Windows); dedicated scripting language "MouseScript" to manage the Mouse, MouseScripts on-demand from HID backdoor shell Trigger payloads via target keyboard No Hardware based: LEDs for CAPSLOCK/SCROLLLOCK and NUMLOCK are scan back and used to branch or trigger payloads (see hid_keyboard2.txt payload) Interactive Hack Script execution Not supported, HID backdoor might be wont to hearth scripts on-demand (via WiFi, Bluetooth or from net using the HID remote backdoor) USB configuration changable throughout runtime supported will perhaps be implemented Support for Hack tool payloads

Support for piping command output to HID keyboard out no supported Switchable payloads Hardware switch manually in interactive mode (Hardware switch might be soldered, script support could be a low priority ToDo. at least

until someone prints a housing for the Pi that has such a switch and PIN connectors)

Interactive Login with show out SSH / serial SSH / serial / stand-alone (USB OTG + HDMI)

III. PROPOSED AUTONOMOUS HACKING TOOL

A. Introduction

The proposed system consists highly customizable USB attack platform, by means of low cost Raspberry Pi Zero or Raspberry Pi Zero W.. The entire unit is portable and can be used on any computer without any external power supply.

We are going to use Python for web app penetration testing and perform Username harvesting, Interception proxies, Command Injection using a Raspberry pi. Unlike the traditional method involving the human coder we are going to automate the test pen starting from network discovery to the vulnerability analysis. The device will be self-sustaining battery bowered and can run autonomously after connecting to the network and send the results to email once the penetration testing is complete.

IV. RESULT AND DISCUSSION

A. Introduction Payload:

HID covert channel backdoor (Pi Zero W only)
HID backdoor features

- Payload to bridge an Airgap target, by relaying a shell over raw HID and supply it from Rakesh Hack Portal via wireless fidelity
- Plug and Play install of HID device on Windows (tested on Windows 7 and Windows 10)
- Covert channel based on raw HID
- Pure in memory, multi stage payload - nothing is written to disk, tiny footprint (compared to typical PowerShell IOCs)
- RAT like control server with custom shell:
- Auto completion for core commands
- Send keystrokes on demand
- Execute Hack Scripts (menu driven)
- Trigger remote backdoor to point out HID covert channel
- Creation of multiple remote processes (only with covert channel connection)

B. Autonomous Hacking tool

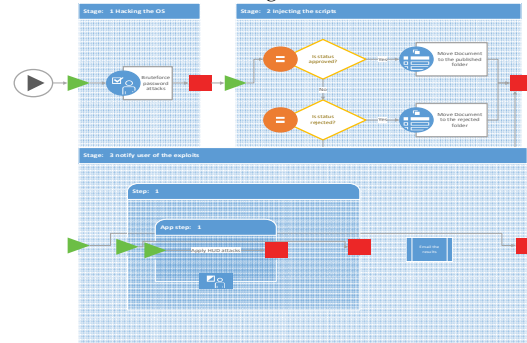


Fig.2.

- console interaction with managed remote processes (only with covert channel connection)
- auto kill of remote payload on disconnect
- shell command to make remote shell (only with covert channel connection)
- server may be accessed with SSH via wireless fidelity once the hid_backdoor.txt payload is running

B. HID backdoor attack chain and usage

1. Preparation

- Choose the hid_backdoor.txt payload in setup.cfg (using the interactive USB OTG mode or one in every of the payloads with SSH network access, like network_only.txt)
- Attach Rakesh Hack Portal to the target host (Windows 7 to 10)

2. Access the Rakesh Hack Portal backdoor shell

- During boot up, Rakesh Hack Portal opens a wireless network referred to as Rakesh Hack Portal (password: MaMe82-Rakesh Hack Portal)
- Connect to the network and SSH in with pi@172.24.0.1
- If everything went fine, you must be greeted by the interactive Rakesh Hack Portal backdoor shell. The SSH password is that the password of the user pi, that is raspberry within the default configuration.

3. Ad-Hoc keyboard attacks from Rakesh Hack Portal backdoor shell (without using the covert channel), may be done from here:

- Entering facilitate shows accessible commands
- Use the SetKeyboardLayout to line the keyboard layout in line with your targets language. This step is very important and will continuously be taken initial otherwise most keyboard based mostly attacks fail.

- To print the present keyboard layout use `GetKeyboardLayout`. The default keyboard language for the Rakesh Hack Portal backdoor shell may be modified in `hidtools/backdoor/config.txt`
- Use the `SendKeys` command followed by an American Standard Code for Information Interchange key sequence to send keystrokes to the target
- As you'll notice, the `SendKeys` command is somehow restricted, no control keys may be sent, even a come back is problematic. thus for additional complicated key sequences the `FireHack Scriptt` command involves facilitate.
- `FireHack Scriptt` accepts the title of a script residing within the `Hack Scriptt/` folder. The folder is prefilled with some demo scripts. If you omit the script name behind the `FireHack Scriptt` command, you'll be bestowed with a menu to settle on a script.
If you surprise why one would write a `Hack Scriptt` causing an + solely, you are thinking within therecent world of Hack tool. With Rakesh Hack Portal and its capbility to run Hack Scriptts dynamically, such short scripts are available in handy. If you do not recognize what i am talking concerning run the Rakesh Hack Portal_youtube.duck script and you will recognize wherever scripts like `AltF4_Return.duck` area unit required

4. Hearth stage one of the covert channel payload ('FireStage1' command)

- As we are able to print characters to the target, we are able to remotely execute code. Rakesh Hack Portal uses this capability to sort out a PowerShell script that builds and executes the covert channel communication stack. This attack works in multiple steps:
 - a. Keystrokes are injected to begin a PowerShell session and sort out stage one of the payload depending on how the command `FireStage1` is employed, this happens in numerous flavours. By default a brief stub is executed, that hides the command windows from the user, followed by the stage one main script.
 - b. The stage one main script comes in two fashions:
 - type 1: A pure PowerShell script that is short and therefore fast, however uses the infamous `IEX` command (this command has the aptitude to create threat

hunters and blue teamers happy). this can be the default stage1 payload.

type 2: A dot internet assembly, that is loaded and executed via PowerShell. This stage one payload executes for longer time, as additional characters are required. But, as you will already recognize, it does not use the `IEX` command.

- It is value mentioning, that the PowerShell session is started without command line arguments, thus there is nothing that triggers detection mechanisms for malicious command lines. Theres no parameter like `-exec bypass`, `-enc`, `-NoProfile` or `hidden ...` nothing suspicious! The disadvantage is, that we want to wait until the PowerShell window opens before writing is sustained. As we aren't able to discover for input readiness associated there are boxes thattake years to point out an interactive PowerShell window, the delay between running `powershell.exe` and beginning of stage1 typeout may be modified with the second parameter to the `FireStage1` command (default is one thousandmilliseconds).
- Last however not least, if you append `nohide` to the tip of the `FireStage1` command line, the Window hiding stub is not dead in direct and you should be able to see all my `sh**ty` debug output.

5. Loading stage 2

- There's no rocket sience here. The stage one payload initializes the essential interface to the custom HID device and receives stage two totally machine-controlled. Stage two includes all the protocol layers and also the final backdoor. It gets directly loaded into memory as dot internet assembly.
- So why dot internet ? the early versions of the backdoor are totally developed in PowerShell. This resulted during amassive mess when it involves multi threading, PS 2.0 compatability while not class inheritance and multi thread debugging with ISE. i do not need to mention that's not possible (if you watched the commit history, there is the proof that it's possible), however there is no profit. To be precise, there are disadvantages: much more code is required to achieve a similar, the code is slower and PowerShell Module logging would be able to catch each single script command from the payload. In distinction to employing a dot internet assembly, wherever the sole PowerShell commands that may get logged, are those that load the assembly and run the stage 2 trigger. Everything is gone

as soon as the payload quits. So ... tiny footprint, yeah.

- But do not get "PowerShell inline assemblies" compiled to a short lived file on disc ???! yes, they do! a minimum of if they are written with CSharp inline code. fortunately Rakesh Hack Portal does not do that. The assemblies are shipped pre-compiled.

6. Using the backdoor connection

- After stage 2 has successfully ran, the prompt of the Rakesh Hack Portal backdoor shell ought to indicate a client affiliation.
- From here on, Rakesh Hack Portal shell commands are usable run facilitate. HID backdoor attack – summary

1. Select hid_backdoor.txt payload
2. Connect Rakesh Hack Portal device to Windows target
3. Hook up with the new spawned Rakesh Hack Portal wireless fidelity with a distinct device (could be a smartphone, as long as a SSH client is installed)
4. Set the right target keyboard layout with SetKeyboardLayout (or alter hidtools/backdoor/config.txt)
5. On the Rakesh Hack Portal shell run SendKeys or FireHack Scriptt to inject key strokes
6. to fire up the covert channel HID backdoor, issue the command FireStage1
7. when the target connected back, enter shell to form a distant shell through the covert channel

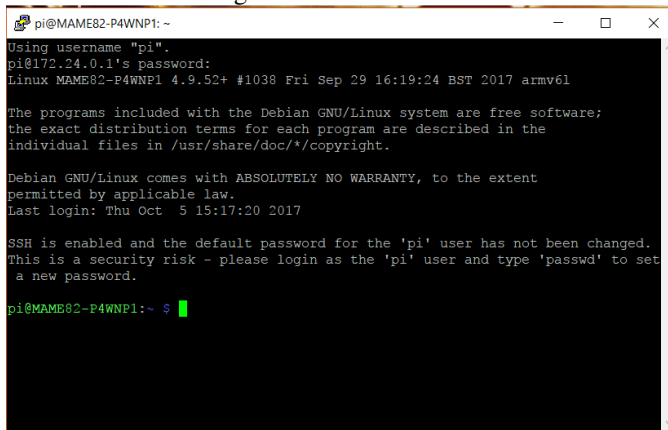


Fig.3.

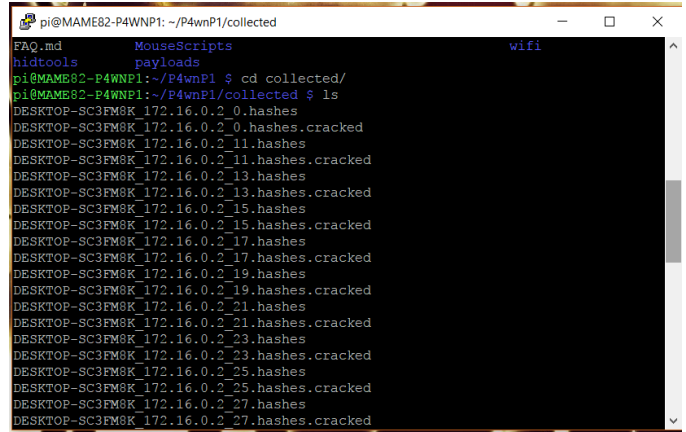


Fig.4.

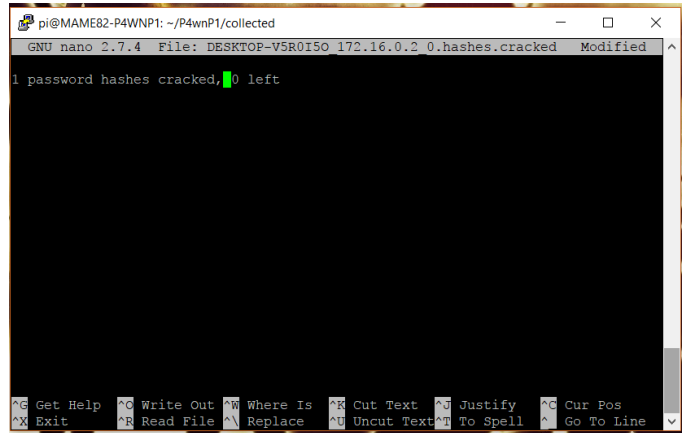


Fig.5.

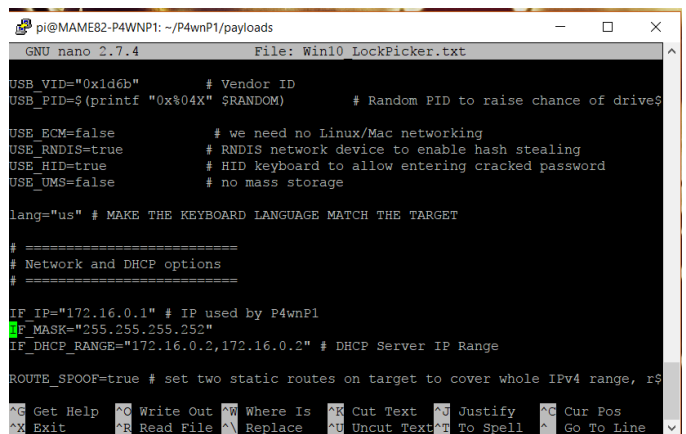
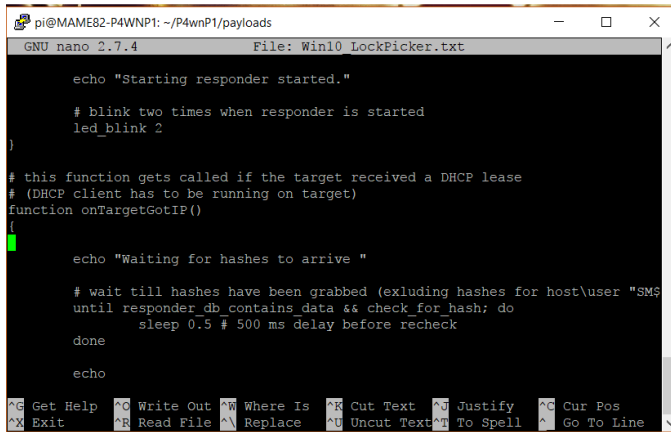


Fig.6.



```
pi@MAME82-P4WNP1: ~/P4wnP1/payloads
GNU nano 2.7.4 File: Win10 LockPicker.txt

echo "Starting responder started."

# blink two times when responder is started
led_blink 2

# this function gets called if the target received a DHCP lease
# (DHCP client has to be running on target)
function onTargetGotIP()

echo "Waiting for hashes to arrive "

# wait till hashes have been grabbed (exluding hashes for host\user "SM9
until responder_db_contains_data && check_for_hash; do
    sleep 0.5 # 500 ms delay before recheck
done

echo

⌘ Get Help  ⌘ Write Out  ⌘ Where Is  ⌘ Cut Text  ⌘ Justify  ⌘ Cur Pos
⌘ Exit      ⌘ Read File  ⌘ Replace  ⌘ Uncut Text ⌘ To Spell  ⌘ Go To Line
```

Fig.7.

REFERENCE

- [1] Kamran Shaukat, Amber Faisal, Rabia Masood, "Security quality assurance through penetration testing", IEEE Xplorer, 2017.
- [2] Boyu Hou, Kai Qian, Lei Li, "MongoDB NoSQL Injection Analysis and Detection", IEEE Xplorer, 2016.
- [3] Manju Khari, Sonam, Vaishali, "Comprehensive study of web application attacks and classification", IEEE Xplorer, 2016.
- [4] Prashant S.Shinde, Shrikant B.Ardhapurkar, "Cyber Security analysis using vulnerability assessment and penetration testing", IEEE Xplorer, 2016.
- [5] E.Danny Alvarez, B.Daniel Correa, I.Fernando Arango, "An analysis of XSS, CSRF and SQL injection in colombian software and website development", IEEE Xplorer, 2016.
- [6] Aldo Hernandez, Victor Sanchez, Gabriel Sanchez, "Security attack prediction based on user sentiment analysis of twitter data", IEEE Xplorer, 2016.
- [7] Z.Ghanbari, Y.Rahmani, H.Ghaffarian, "Comparative approach to web application firewalls", 2016.
- [8] Deepak Kshirsagar, Sandeep Kumar, Lalit Purohit "Exploring usage of ontology for HTTP response splitting attack", IEEE Xplorer, 2016.
- [9] Prasad V.Kalne, Vaishali L.Kolhe, "A new avatar dynamic image based CAPTCHA service on cloud For Mobile devices" IEEE Xplorer, 2015.