PREVIEW

HAKIN9

PHYSICAL PROTECTION

IT SECURITY MAGAZINE

VOL.16, NO. 09

IOT AND RASPBERRY PI HACKER'S TOOLKIT

Hacking IoT with IoT

Device and Network Based Attacks

IoT Bug Bounty 101

Raspberry Pi "Evil Server"

AND MORE...



TEAM

Editor-in-Chief
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Managing Editor

Magdalena Jarzębska
magdalena.jarzebska@hakin9.org

Editors:

Marta Sienicka
sienicka.marta@hakin9.com

Marta Strzelec
marta.strzelec@eforensicsmag.com

Bartek Adach
bartek.adach@pentestmag.com

Michalina Szpyrka
michalina.szpyrka@eforensicsmag.com

Proofreader:
Lee McKenzie

Senior Consultant/Publisher:
Paweł Marciniak

CEO:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

Marketing Director:
Joanna Kretowicz
joanna.kretowicz@eforensicsmag.com

DTP
Magdalena Jarzębska
magdalena.jarzebska@hakin9.org

Cover Design
Hiep Nguyen Duc
Joanna Kretowicz

Publisher
Hakin9 Media Sp. z o.o.
02-676 Warszawa
ul. Bielawska 6/19
Phone: 1 917 338 3631

www.hakin9.org

All trademarks, trade names, or logos mentioned or used are the property of their respective owners. The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

BETATESTERS & PROOFREADERS

Lee McKenzie

Hammad Arshed

Avi Benchimol

Amit Chugh

Kevin Goosie

Craig Thornton

Paul Mellen

Alex Giles

Alexandre D'Hondt

David Molik

Ivan Burke

Olivier Caleff

Gilles Lami

Nasreddine Bencherchali

Daniel Sligar

Al Alkoraishi

Jay Kay

Christopher Pedersen

Dear readers,

The presence of IoT devices in our lives is getting more and more visible each day. Watches, lights, cars, cameras, doorbells, locks - every smart device is a potential target of a hacker's attack. That is why we decided to dedicate our September issue to the topic of IoT hacking and to focus on a more offensive approach, using the fan favourite Raspberry Pi! Let's dive into it!

We're starting off with a great article *Hacking IoT with IoT*, in which Dan Dieterle explains how to attack an office building security camera system with Raspberry Pi. Later on we have *IoT Bug Bounty 101* - this article can serve as a guide for those of you, who want to start penetration testing on IoT devices.

In *Device and Network Based Attacks* the author explains and analyses various kinds of attack scenarios, targeted toward IoT devices. The next article, *Raspberry Pi "Evil Server"*, is a tutorial on how to install an evil server on a Raspberry Pi and how to perform different forms of cyber attacks with it.

In *Security Vulnerabilities in Using Home IoT Devices During Pandemic Period* you'll learn how the pandemic influenced IoT security and what are the newest types of IoT vulnerabilities that emerged during that time. Coming back to the topic of Raspberry Pi, in *Pi-Injector* you can learn how to implement this tool, which is used for keystroke injection in wireless mice and keyboards.

Later on, in *How I Learned That I Should Worry About my Connected Devices and Not Just the Morning Before a Pentest*, one of our authors explains how important it is to prepare your smart device for pentesting, and how to do it properly. Then, in *CVE-2021-33056 Linphone SIP Protocol Stack*, the author covers a case of one of the recently discovered Linphone SIP vulnerabilities.

If you'd like to have a little bit more variety, the last two articles are for you! One of them is a beginner-friendly tutorial on *Network Vulnerability Scan Using OpenVAS*. The second one will teach you how to *Crack Password-Protected Microsoft Office Documents*.

As you can see, this issue is full of quality articles. We hope you enjoy them, and make good use of the knowledge they are providing.

We would like to thank our amazing contributors, reviewers, and proofreaders, without whom this issue wouldn't be possible.

Stay safe and enjoy!

Magdalena Jarzębska and Hakin9 Editorial Team

Contents



Hacking IoT with IoT

by Daniel W. Dieterle



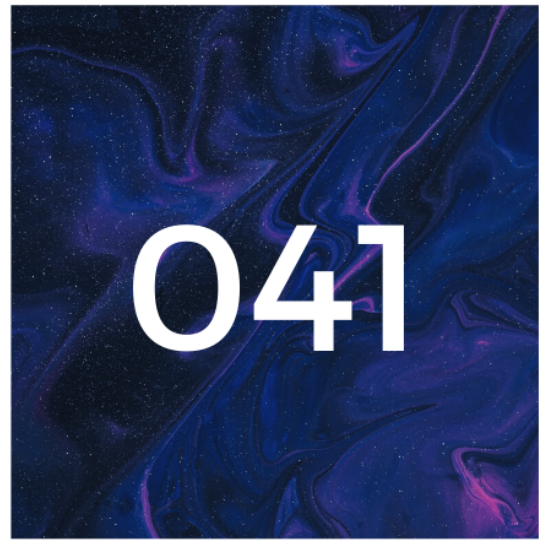
IoT Bug Bounty 101: Hacking the Internet of Things From a Beginner Point of View

by Mark Antwi Acquaisie



Device and Network Based Attacks

by Mukul Kantiwal



Raspberry Pi “Evil Server”

by Atlas Stark

Contents



Security Vulnerabilities in Using Home IoT Devices During Pandemic Period. A Case Study on Personal and Business Impacts

by Lochana Koralage



Pi-Injector

by Mayank Tanna



How I Learned That I Should Worry About My Connected Devices and Not Just the Morning Before a Pentest

by Jean-Georges Valle



CVE-2021-33056 Linphone SIP Protocole Stack

by Felipe Hifram

Contents



Network Vulnerability Scan Using OpenVAS

by Matheus Rebouças



Crack Password Protected Microsoft Office Documents

by Dr. Akashdeep Bhardwaj

HACKING IOT WITH IOT





DANIEL W. DIETERLE

Daniel W. Dieterle, aka "CyberArms", has been in the computer industry for over 20 years, and has worked as a security author, researcher & consultant. He has authored six books based on Kali Linux, and is currently working on his seventh, "Advanced Security Testing with Kali Linux". Daniel also runs two tech blogs - cyberarms.wordpress.com & DanTheIOTMan.com.

IoT (Internet of Things) vs IOT - reminiscent of the old Mad Magazine “Spy vs Spy” cartoon where there were two identical-looking cartoon spies of different colors that were always trying to kill each other. The rise of vulnerable deployed IoT devices and the offensive use of IoT devices is skyrocketing. In this article, we will cover attacking an IoT device, an office building security camera system, with another IoT device, a Raspberry Pi.



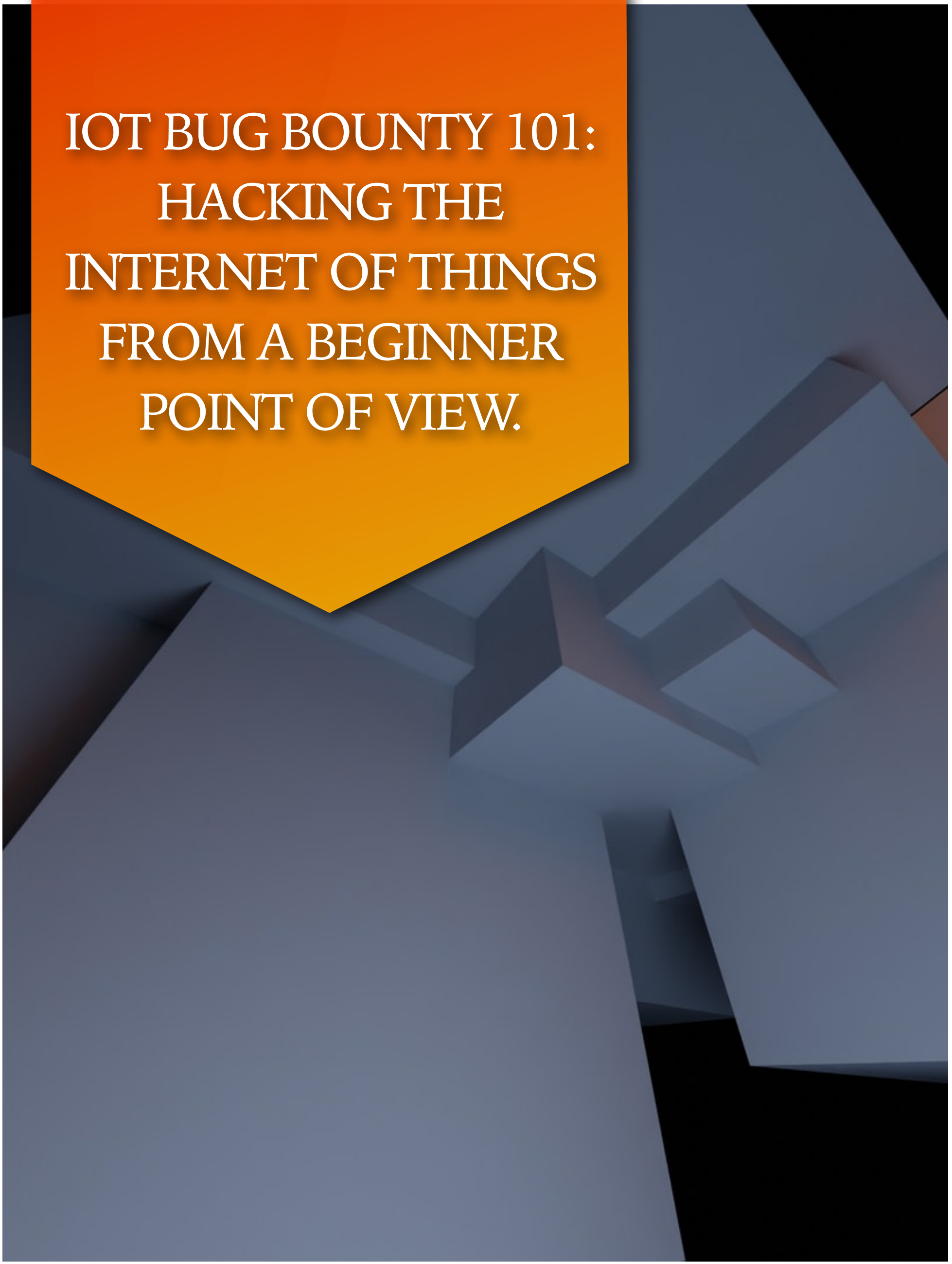
IoT Devices - What are they?

An IoT device is a physical device with intelligence (sensors, data collection, monitoring) that communicates with other devices over the network. IoT devices include security systems, smart appliances (online TVs, refrigerators, coffee makers, etc.), building and machine control, monitoring devices, scanners, and sensor arrays that are accessible over the Internet. Many “Maker” boards like the Raspberry Pi and Arduino are also used frequently in IoT applications. For this article, we will specifically focus on the Raspberry Pi and a Smart Camera security system.

Vulnerable Systems - What People Don’t Understand

IoT devices are so popular because they bring online connectivity and monitoring to almost every industry. The large problem, and what people don’t understand, is that there is a mini web server hiding inside them, and it is usually Linux based. Once an IoT device is deployed, many times it gets forgotten. It just sits on the wire, collecting or monitoring, and not usually getting updated. Security monitoring IoT devices are still fairly uncommon in the business world, and as we say jokingly in the Red Team world, “they don’t make anti-virus for refrigerators”. This makes them a prime target for unethical hackers.

IOT BUG BOUNTY 101: HACKING THE INTERNET OF THINGS FROM A BEGINNER POINT OF VIEW.





MARK ANTWI ACQUAISIE

Mark Antwi Acquaisie is a cyber security professional and an internet-of-things (I.O.T) enthusiast. Mark is a Senior Application Specialist at eProcess International S.A.(ECOBANK) over four years.

With an exemplary entrepreneurial skill, Mark Co-founded Tenerdz Company Ltd. A member of OWASP, member of Institute of Electrical and Electronics Engineers (IEEE), member of Internet Society (ISOC Ghana chapter), Executive Member of Arduino (Genuino) Ghana and volunteers in a few NGO's (Autism ambassador's society of Ghana). He was also a Tenerdz robotics instructor at a few Tertiary schools in Accra, Ghana and won gold and bronze with his students from Galaxy International School at the Olympiad in 2017.

Mark is highly skilled in Cybersecurity (Purple Teaming), I.O.T., Cloud Technologies, Programming (Python & React), DevOps and Machine Learning.

As a cyber security professional, Mark holds several IT and security certifications, such as CompTIA Security+, ITIL Certification, ICSI | CNSS Certified Network Security Specialist (CERT), IBM Cybersecurity Compliance Framework & System

Administration (CERT), MITRE ATT&CK, and also pursuing AWS and CISA certifications.

Mark holds a Bachelor of Science in Computer Engineering from Ghana Technology University and currently pursuing a Master of Science in Information Security and Digital Forensics at University of East London.

You can follow Mark on Twitter at @MAcquaisie007

Introduction

The fuss about bug bounties is so fascinating and has been the current cyber security trend for some years now. A bug bounty program as defined on wiki is a deal offered by many websites, organizations, and software developers by which individuals can receive recognition and compensation for reporting bugs, especially those involving security vulnerabilities. These pentests were usually performed on web, mobile and cloud technologies until just recently when some of these bug bounty platforms introduced bug bounties on IoT technologies and smart devices. Even though this category is new, it has grown enormously over the years. IoT bug bounty compared to the other categories has a more flexible entry barrier, targets can be easily found because of performing very easy tasks, it's less competitive as compared to the other categories and due to the wide market demand of IoT devices, manufacturers keep on making unsecured products.

You might be wondering, what does it take, where do I go and how do I join this fun but educative and adventurous journey that comes with a reward? How cool is that? This article will serve as a guide for persons wanting to start penetration testing on IoT devices.

Prerequisite Skills Needed

To be able to perform adequately and be familiar with some terminologies used in pentesting these devices you must be conversant with the ***BASIC*** knowledge of the skills listed below.

☑ Internet, HTTP, TCP/IP

☑ Networking

☑ Command-line

☑ Linux

☑ Web and mobile technologies, JavaScript, PHP, Java

☑ At least two programming languages preferably (Python/C or C++)

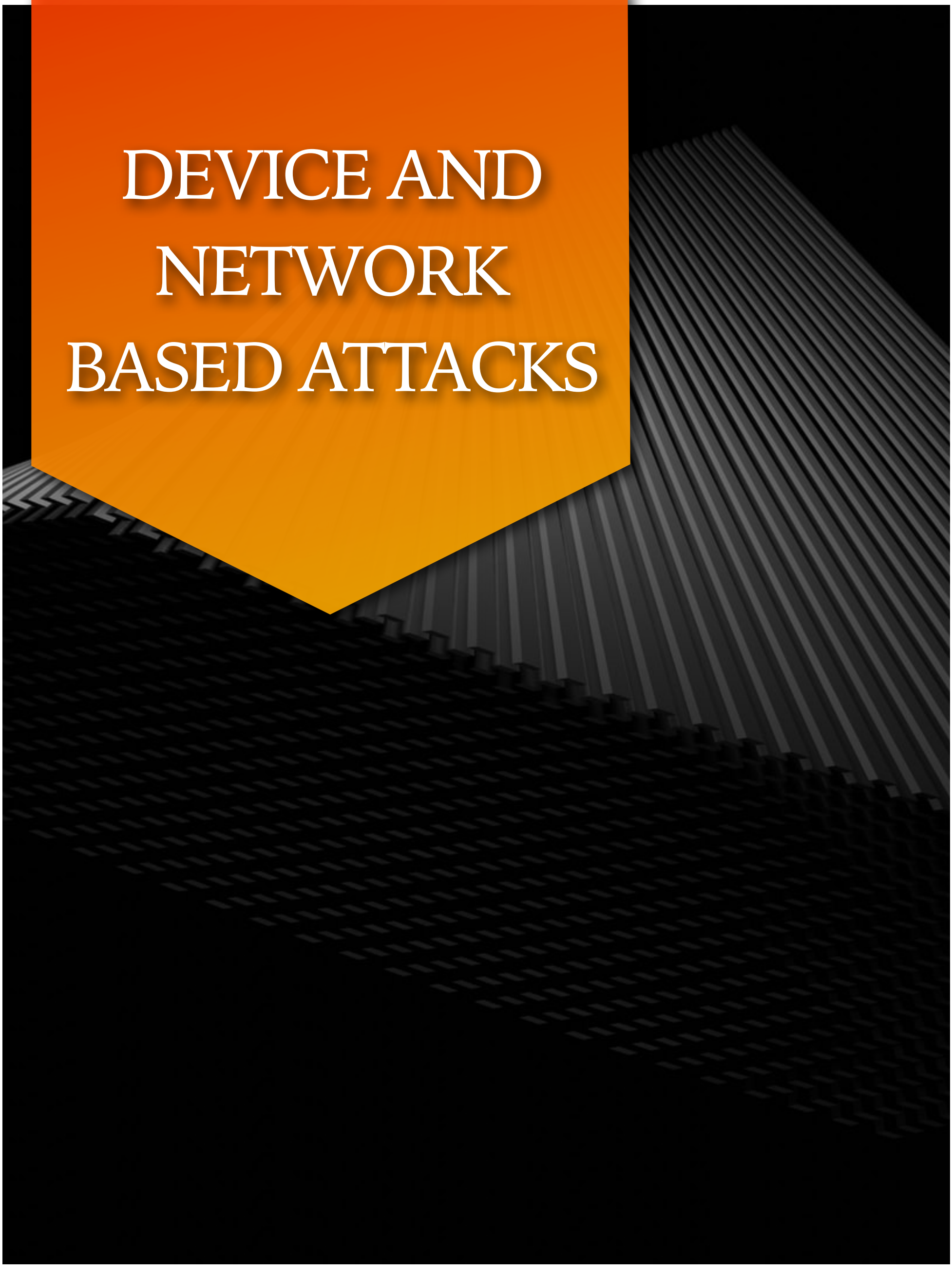
☑ OWASP IoT Top 10

☑ Microcontrollers (Arduino)

Understanding the IoT Architecture

A quote by Sun Tzu once said, if you know the enemy (IoT Architecture) and know yourself (basic skills above), you need not fear the result of a hundred battles (IoT bug bounty programs). Would you agree with me? What could be the difference between a garden sprinkler and an IoT-connected garden sprinkler that does the same thing? In one word: I would say "data". An IoT-connected garden sprinkler device would have its actuators turn on the device after receiving data from its

DEVICE AND NETWORK BASED ATTACKS





MUKUL KANTIWAL

My name is Mukul Kantiwal and I work as a Security Analyst for a privately owned organization. I am the instructor for web application attacks and API hacking. I am also listed in the top 100 hackers by ringzer0team. I have more than 5 years of experience in the field of Information Security, and I have worked on Web Apps, Mobile Apps, APIs, networks, Thick client, RFID, etc. I have recently started a YouTube channel where I am planning to show different types of techniques and post some tutorials about web application security. Feel free to reach out if you have a topic in mind on which I should prepare some content.

Apart from researching things in the cyber security domain, I write poetry sometimes and I have a small poetry blog named poetground. If any of the readers are interested in posting any of their poetry, do reach out to me for poetry as well or you can use the poetry to impress your girlfriend/wife ;)

Email: keanureeves456@gmail.com

LinkedIn: https://www.linkedin.com/in/mukul-kantiwal

Github: https://github.com/mukkul007

Youtube channel: (https://www.youtube.com/channel/UCqU-tXIoRnxcjzYxtbDzdtw) The Cyber Ground

Poetry: https://www.poetground.com/

Introduction

IoT stands for Internet of Things. The Internet of Things (IoT) is the network of devices and components that are connected to each other or the internet. These devices serve different purposes, from monitoring the heart rate of a person in the form of wearable devices to making a whole city “smart” in the form of various sensors. These days, one can find these devices nearly everywhere. We are surrounded by them. With the rapid increase in the number of these devices everywhere, from our offices to our homes, it is important to make sure these devices are doing only their work, and not spying on us or taking commands from anyone else.

Description

IoT security is an important field that spreads over device security, network security and a lot more. To dive into IoT security, one should have a basic knowledge of the Linux operating system and basic knowledge of computer networks. The reason is that these devices communicate over the network and are majorly created by using the components of Linux-based operating systems.

There are many things that can be covered under IoT security but to understand the attacks or craft an attack over IoT devices, one should have a little information over three major sections:

1. Bootloader - The bootloader or boot program or bootstrap loader is a special software program that is responsible for locating and loading all the required files, e.g., kernel, filesystem, and starting the operating system. Universal Boot Loader (U-Boot) is one of the most popular open-source bootloaders that is used in a lot of different architectures/platforms/devices.

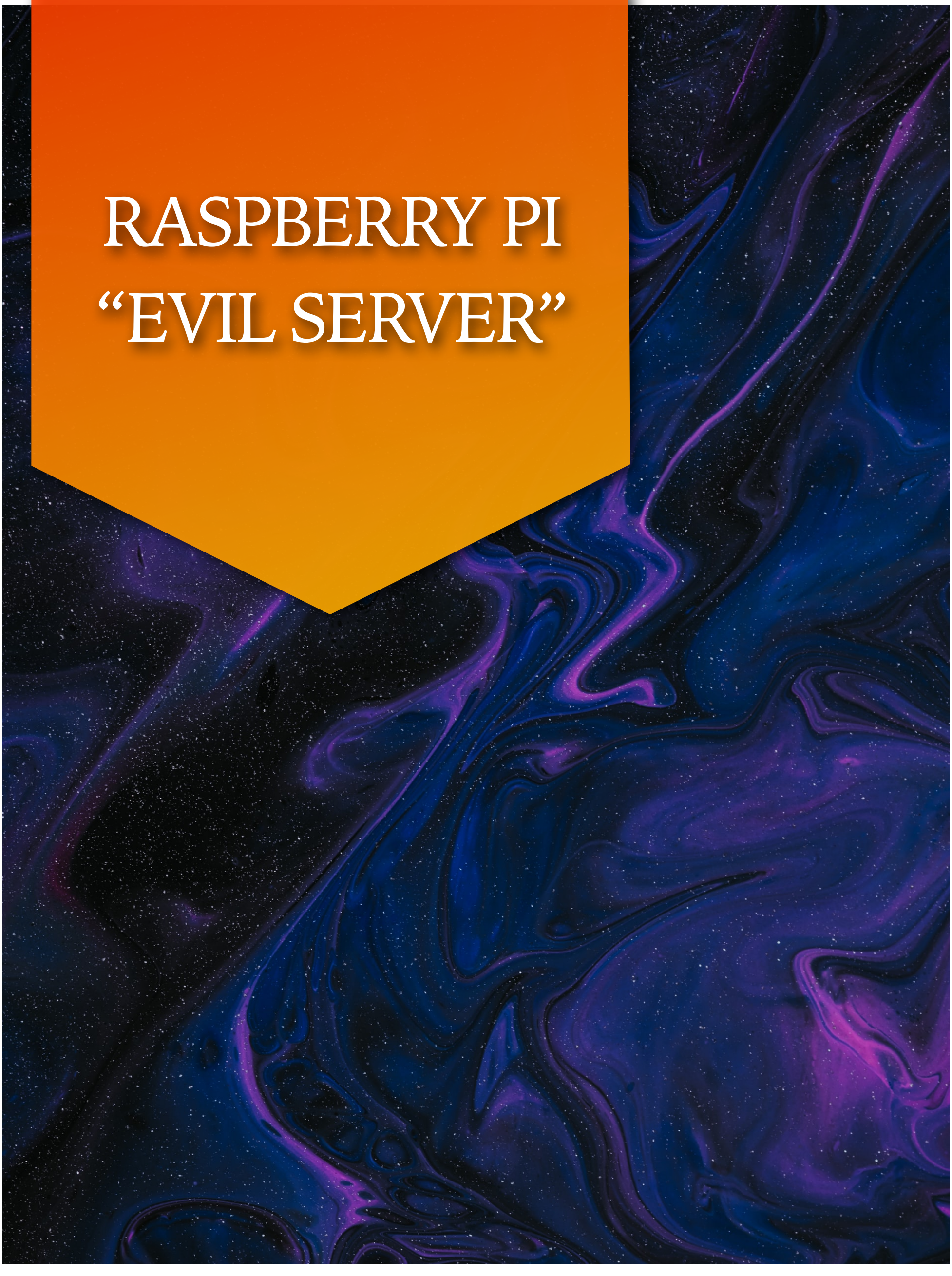
2. MQTT - MQ Telemetry Transport (MQTT) is a lightweight protocol that is widely used by IoT devices to communicate with their brokers/servers situated on remote locations and clouds. The protocol is simple and consumes much less bandwidth and power than its peers, making it best for devices where network bandwidth and power are constrained. It is a two-way communication protocol. MQTT works in a subscribe-publish model. Brokers or servers play middleman for the publishers (producers, e.g., sensors) and subscribers (consumers, e.g., central monitoring system).

3. AQMP - Advanced Message Queuing Protocol (AMQP) is a standard protocol for passing messages between applications. It is mostly used in business-related applications and provides buffer capacity for incoming/outgoing messages. However, with IoT devices entering the enterprise space, the AMQP implementations like RabbitMQ are being used to send/receive/buffer the messages.

Boot a machine from U-boot and SD card

Let us take up a scenario and see how an ARM vexpress emulated machine is supposed to boot from an SD card (SD card is already mounted to the machine). In this scenario, the U-Boot is not able to locate the necessary files (Kernel, DTB) and hence it fails, exposing the U-Boot console to the user.

RASPBERRY PI
“EVIL SERVER”





ATLAS STARK

Atlas Stark is a security researcher at Stark Industries Inc. with 16+ years in the technology industry. Currently providing cyber security solutions and OSINT services to anti-human trafficking non-profits that aid in investigation and victim recovery. He also consults with state level law enforcement agencies concerning hacking related incidents. He splits his time between California and Tennessee.

Q/C: Please email stark@starkinternational.se with any questions or concerns.

Evil Server

At times you may find yourself without additional hardware and resources to perform a viable security audit for any number of reasons; you may not want to pay for AWS or a DigitalOcean droplet, you may need to stay covert, maybe you’re just getting started as a penetration tester, or you’re simply an independent security researcher that likes to have full control over the platform. Whichever bracket you classify yourself in, being able to establish a remote access command and control server will not only enrich your engagement, but will also increase your Raspberry Pi skills. It’s also a whole lot of fun to boot. Get it? Boot.

Why are we calling this an evil server? We are calling it an evil server because once implemented you will be able to do everything that a bad actor can do, such as sniff, capture and analyze a variety of network traffic. Use the platform to inject code and scan network devices for vulnerabilities. Perform man-in-the middle attacks and use an Apache web server to stage websites that will be able to mimic an organization or public site in order to steal credentials or test the workforce of a company. Also, discover any and all computers and devices connected to the target network. You will also be able to install key-loggers, backdoors and other Trojans onto target computers and much, much more. In the links section, I have included a variety of resources that will help guide you to perform tasks such as the ones just listed.

Who is this article for?

A vast assortment has been written about various Raspberry Pi projects and add-ons to date and all of them are fun and useful in their own way. This article is geared for anyone that is interested in getting started with the Raspberry Pi platform in the realm of penetration testing or security research. Even if you are an experienced Pi user, this tutorial may provide you with an additional set of skills that you can implement in your research or accompany you on your next engagement.

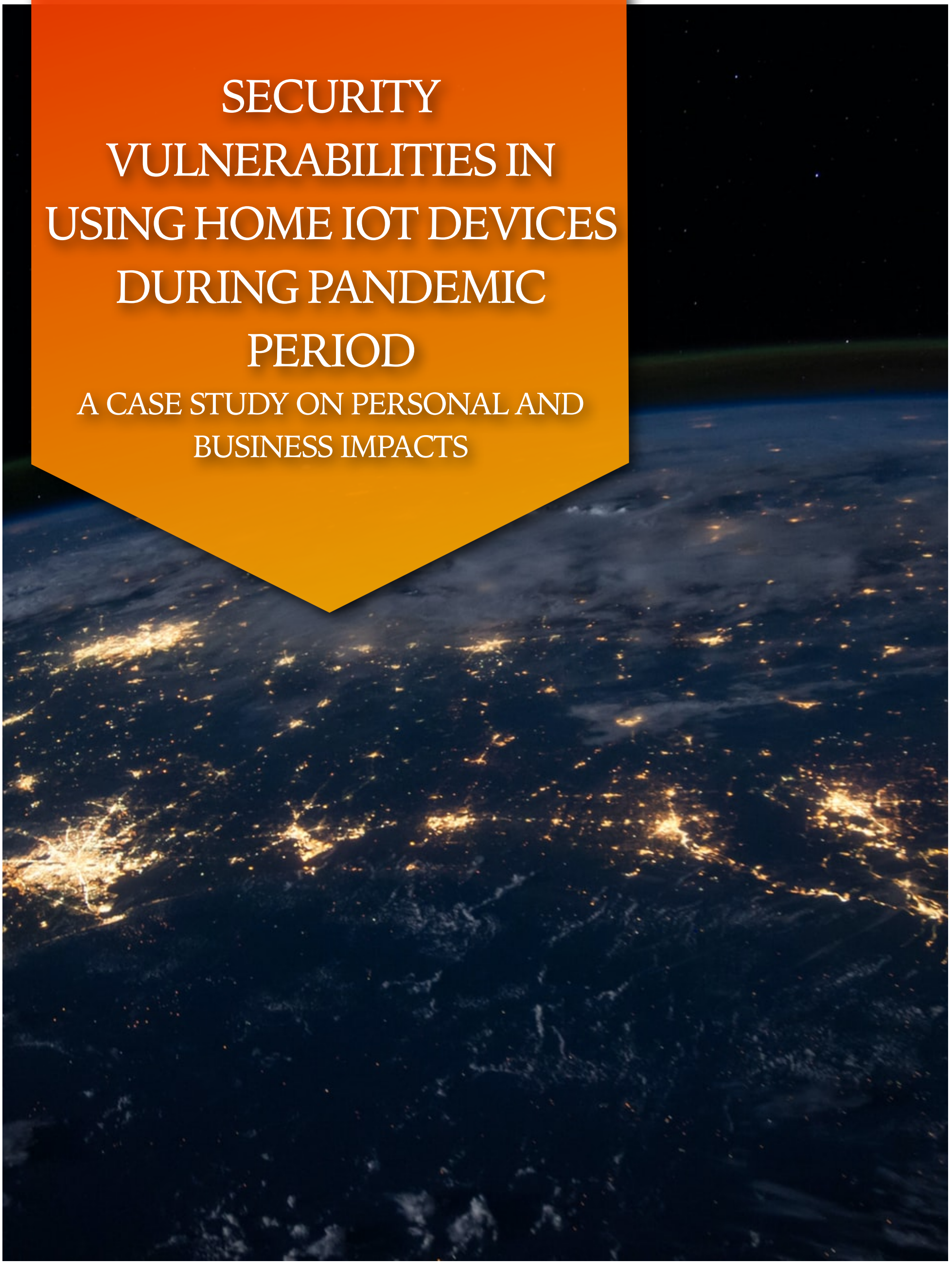
Raspberry Pi

Raspberry Pi, since its commercial debut in February 2012, has become a go-to platform for enthusiasts, hackers and security professionals alike, largely due to its compact size, competitive price and ability to scale to the outside world efficiently and effectively.

Let’s face it, the costs associated with penetration testing and security research can add up quickly. Not only is it the initial sticker shock jolt to your wallet, but the maintenance costs associated with the equipment can be daunting as well, not to mention the amount already spent on education and insurance. If you need a bang for your buck then look no further than the power packed Pi platform with its main feature being an extremely approachable price tag. This credit card sized micro machine packs a serious punch, not to mention its versatility. The Raspberry Pi can be configured as a smart home controller, a game server, a command and control server or just about any IOT functionality you can dream up. As cool as all that sounds, we will be using ours to construct an evil platform with which to conduct testing.

SECURITY VULNERABILITIES IN USING HOME IOT DEVICES DURING PANDEMIC PERIOD

A CASE STUDY ON PERSONAL AND BUSINESS IMPACTS





LOCHANA KORALAGE

Lochana Koralage is an experienced Systems Engineer, currently working in the higher education sector in the Leading Higher Education Institute in Sri Lanka. The author holds an Honors Degree in Computing, and a master's degree, specializing in Cyber Security.

Lochana koralage - MSc IT (specializing in cyber security), BSc (hons) Computing CEH CHFI.

Introduction

During this period of global chaos caused by the pandemic, a 'new norm' has set out around the world. This new norm has resulted in a set of new threats and vulnerabilities, social and cultural impacts, mental and physical health issues, etc. Through this case study, the security impacts set out by the use of home IoT devices, such as baby monitors, are to be discussed in general, and in specific relation to special conditions set out by the pandemic situation, Covid-19, starting at the end of 2019, continuing for almost two years. Thereby, the risks of exposure of business assets to external parties or attackers with the use of home IoT devices have grown too, as a result of home environments and business environments increasingly becoming one environment through the increase of 'working – from – home' as the new norm.

Use of Home IoT Devices

With the evolving use of smart home concepts, IoT (Internet of Things) devices are commonplace in most houses. IoT are smart devices that are connected to the internet to communicate. Even though you see these as single purpose devices, a 'thing' connected to the internet is not just a home appliance, it is a computer that has the ability to communicate with other devices, act as a server or a client, gather and process information as it has a CPU, etc. Therefore, rather than the mere lack of a mouse and a keyboard, any small IoT device you use at home would be a small multi-purpose computer.

A computer is vulnerable, vulnerable to external threats, attackers, etc. There are currently a ton of different IoT devices used in homes for simple purposes. However, according to many experts, it is an accepted truth that there is no 'single purpose computer' created so far. Every IoT device, alongside end-user restrictions, provides you with a general-purpose computer that only performs a certain task in your eyes. However, these systems are more often proven to be flawed in many different ways.

For example, DRM (Digital Rights Management) is one such manufacturer-imposed restriction method for things like limiting the Media Payback properties for some users, while allowing others. Such systems have been the target of hackers for so many years, that they easily gain unauthorized access through preservation efforts and some limited resources available. This may be tough for some users, but some users with technological capabilities often tend to breach these boundaries to access the actual general-purpose computer that is included in the device and hack it. DRM-like systems are mainly used for access control. Most users depend on those systems to protect their data, prevent unauthorized access and delete the data properly. Yet, these systems are often proven to be weak and vulnerable to attacks.

Security in IoT devices

We discussed that IoT devices or home appliances that are connected to the internet, such as baby monitors, are in fact general-purpose computers. Yet, there is one key difference in a general-purpose computer known to you and the devices discussed here. In a normal computer, there is the capability to send security updates, patch updates, and software upgrades, and the user is aware of the threats and vulnerabilities as well as remedies for them up to a certain extent.

In contrast, there are limited upgrades or updates that are done to an IoT device once it leaves the manufacturer. Therefore,

PI-INJECTOR





MAYANK TANNA

Mayank Tanna is a cyber security researcher/student and a certified ethical hacker from India with a keen interest in IoT and Ethical hacking. Currently, he is competing at a national level in the field IoT for a number of projects. In the year 2020, he achieved a score of 97.6% in EC-Council's Certified Ethical Hacker v10 due to which he was nominated for CEH Hall of Fame 2021. Apart from this, Mayank is a bug bounty hunter and has also worked on multiple projects in an effort to fight the menace of rape.

LinkedIn: https://www.linkedin.com/in/mayanktanna/

Overview:

- Introduction
- Hardware Requirements
- Operating System Required
- Theory and attack details
- Installation
- Attack Execution
- Conclusion

Introduction

Pi-Injector is a tool used for keystroke injection in wireless mice and keyboards using Raspberry Pi. This works with certain types of vulnerable mice and keyboards that are still used today. Many companies in the current day and age are using dated wireless mice and keyboards, which enables hackers to easily gain access - which in turn may lead to data breach, privilege escalation, etc.

Hardware Requirements

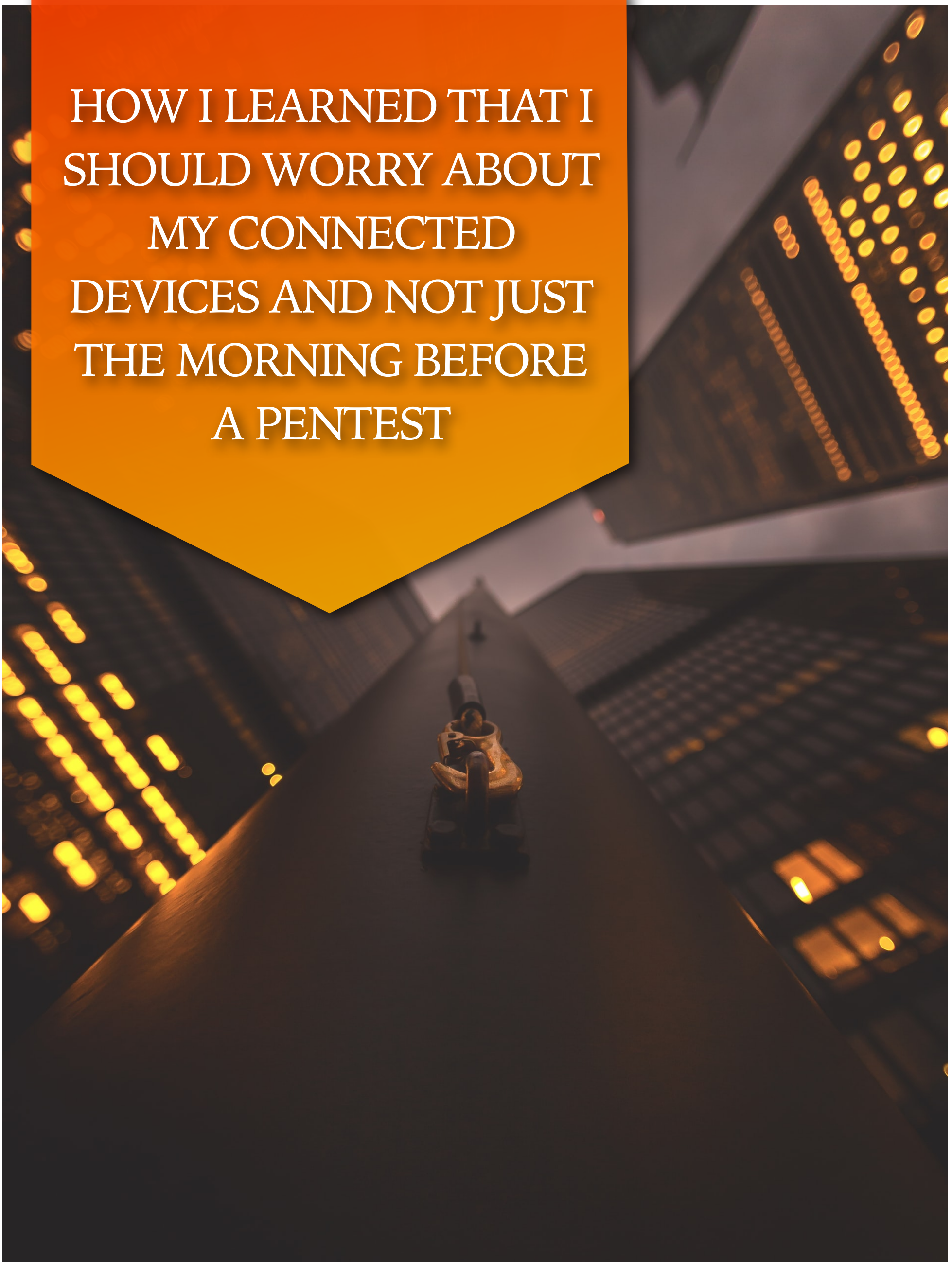
Raspberry Pi:

The Raspberry Pi is a palm sized computer that can plug into a monitor or TV, and uses a standard keyboard and mouse. It is capable of doing everything that a desktop computer can do, from browsing the internet and playing HD videos, to making spreadsheets, word-processing, and playing games. It is widely used in IoT projects.



Image 1: Raspberry Pi

HOW I LEARNED THAT I SHOULD WORRY ABOUT MY CONNECTED DEVICES AND NOT JUST THE MORNING BEFORE A PENTEST





JEAN-GEORGES VALLE

Jean-Georges Valle is a hardware penetration tester based in Belgium. His background was in software security, with hardware being a hobby, when he started to look into the security aspects of hardware. He has spent the last decade testing various systems, from industrial logic controllers to city-scale IoT, and from media distribution to power metering. He has learned to attack embedded systems and to leverage them against cloudscale infrastructure. He is the lead hardware technical expert in an PwC Belgium's offensive security team.

Jean-Georges holds a master's degree in information security and focuses on security at the point of intersection with hardware and software hardware and software interaction, exploit development in embedded systems, and open source hardware.

A common question is shared by both the young and ruthless pentester preparing for a connected or embedded device test and the (probably hired last minute) IoT Security analyst, shot into a project that has to hit the market in two weeks, while having to face a firmware development team that discreetly tests if crosses and holy water reveals the demon (Robin from testing "involuntarily" dropping a glass of "totally mundane" water on you this morning? You bet...) they are sure he is.

This question is "Where do I start?"

The only general answer to this that makes any sense is... "It depends?"

There is a wonderful land where legendary "*Security requirements*" are discussed in advance and written on a fabled "*documentation*", where "*threat modeling*" is discussed by wise wizards and witches at least a few months before release, where "*agile*" doesn't have programmers running in all directions like headless chickens because Alex from marketing had a "brilliant" idea yesterday...

If you find this place, please email me... I've never been and I heard it's nice.

In the meantime, you have an unknown system to test or to prepare for release. Whether you'd like to pwn the whole infrastructure for bragging rights during conference parties or to prevent the pwnage in order to keep a decent wage coming your way (yes, that's less glamorous but the bills won't pay themselves)...you have to start identifying the same components. Only the intent (to compromise or to safeguard) changes. Either way these are the client's nuts and they are yours to "steal" or to protect...in order to keep that in mind, we'll call our client (or our boss) Scrat.

1) Understanding the device's business

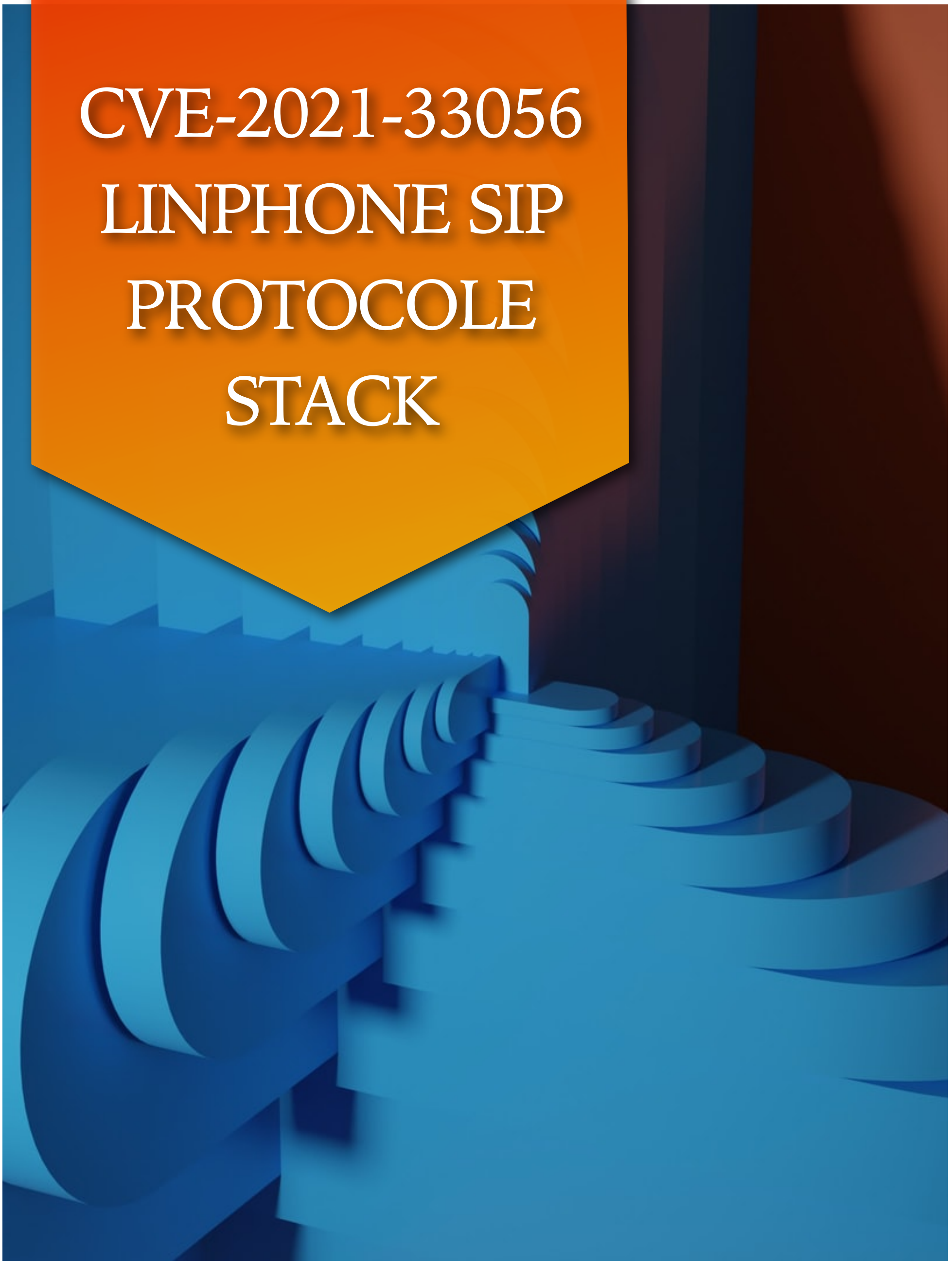
First things first. What is the thing supposed to be doing as its primary function? A cloud-scale calf monitor with IA based meat quality prediction, an air quality monitor that is shared across the students of a school or a peta-bit fiber SDN router/firewall/nextgen threat prevention solution that also makes coffee? All our shiny new toys usually come with a main function and a plethora of ancillary features (along the lines of cloud-connected, recipe saving, coffee order triggering, AI based behavioural analytics, blockchain something-something scams in general).

List all of these in writing, from the user manual, or any piece of documentation you can put your grubby hands on or from interviewing Scrat (and his team). Each of these functions will either drive or be supported by the security functions (or properties) of the device.

For each of these functions, formally identify:

- ☑ The functionalities, whether accessible by the user (making coffee, driving around, etc.) or by the vendor (remote maintenance, update, etc.).
- ☑ The (potential) third parties involved (platform providers, software and hardware components vendors, analytics providers, etc.). They are present for a reason and could be a way in or a stakeholder in a targeted process.

CVE-2021-33056 LINPHONE SIP PROTOCOLE STACK





FELIPE HIFRAM

Former social engineering professional, he focuses on good habits of use and privacy on the internet.

He has worked for companies in Brazil, Germany, Ukraine, Oman and Bahrain.

Today, in addition to writing articles, he works as an Art Director and is a Fashion Design student.

SIP

For starters, in case you are not familiar with SIP, it is a protocol whose abbreviation means "Session Initiation Protocol", it is a fundamental basis in the communication of systems (including IoT) that use video and voice, and is very present in companies of different sectors.

Widely used in security systems, such as more modern cameras and doorbells, as reported by Claroty itself, the company responsible for discovering the vulnerability described in this article.

In more depth, the SIP protocol is used at startup and shutdown, in addition to being responsible for maintaining and modifying sessions in real time for two or more IP terminals, involving voice, video and messages, as mentioned above.

The protocol has a textual basis similar to HTTP and has two different versions, the first one uses a method to define the origin of the request, and there's also a URI (Uniform Resource Identifier) of the same request indicating where it should be sent.

Linphone

In this project, this is an open-source application, the first to use SIP on Linux, which uses the protocol as a basis for developing voice, video and messaging communications.

The main difference is the use of IP networks to make VoIP calls, and also messages, which already use SIP.

On its official website, Linphone claims the presence of more than 200 clients from the corporate sector, involving major brands such as Acer and Swisscom, and others from different sectors (automation, residential, IoT, etc.).

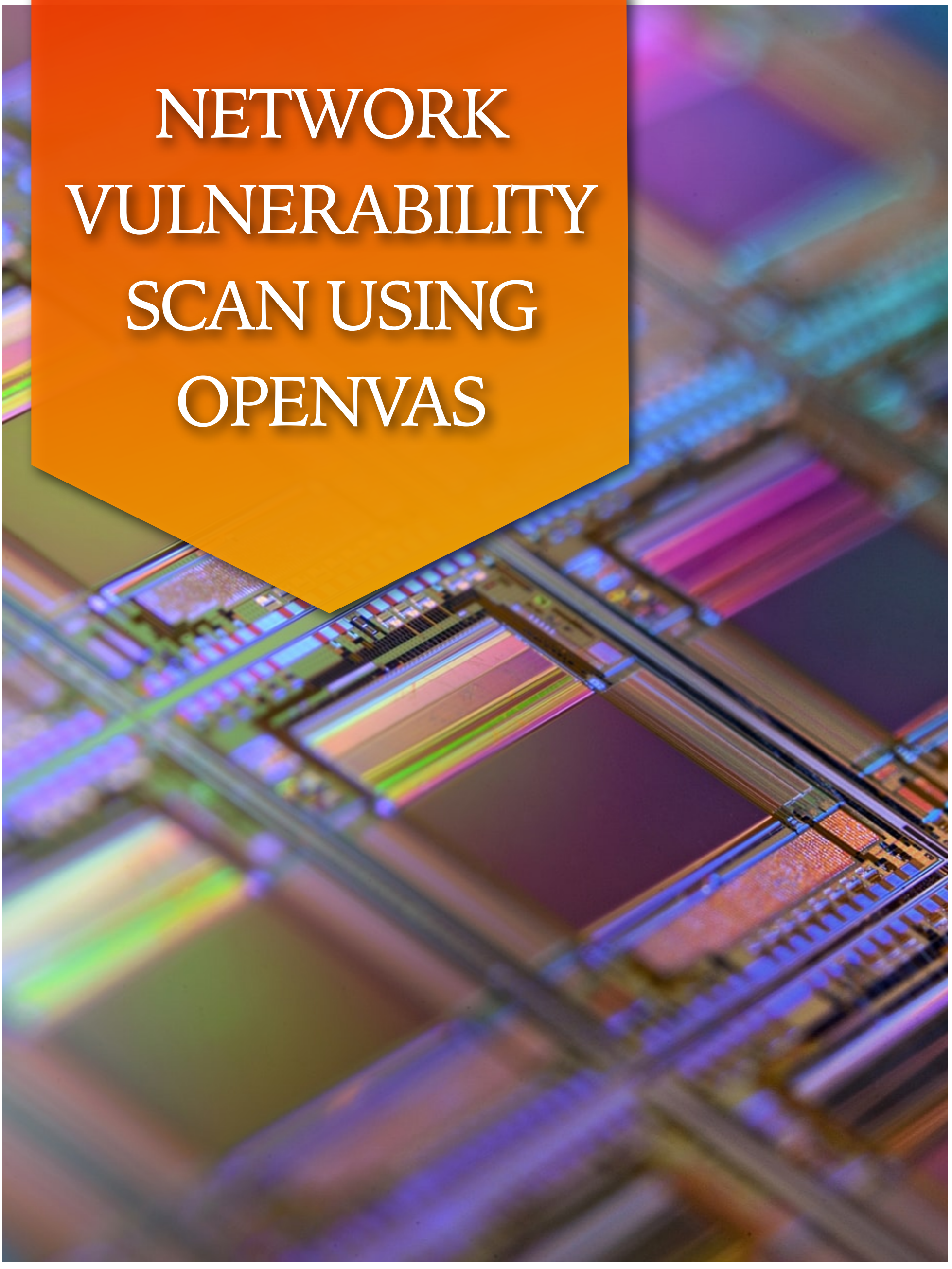
Discovery

A team of Claroty professionals, described as "Team82", discovered a flaw in a "belle-sip" component of Linphone, which is a vulnerability characterized as "null pointer dereference".

The "belle-sip" C library contains an object-oriented API and, in addition to having an HTTP/HTTPS client implementation, it is used in SIP transport, that is, transactions and conversation layers.

The vulnerability would eventually allow the service to crash through a denial of service, which in turn could be caused by sending an invalid SIP header ("zero click" attack), without requiring any action from the victim, and can be executed 100% remotely.

NETWORK VULNERABILITY SCAN USING OPENVAS





MATHEUS REBOUÇAS

My name is Matheus Rebouças, I'm a computer science student, CTF player and security enthusiast. Feel free to check me out on LinkedIn for any doubts, critics, or just hang out.

Introduction

OpenVAS is an open-source vulnerability scanner developed by Greenbone Networks, which can perform unauthenticated as well as authenticated scans. That is, scans with and without the usage of credentials. For this, you will need to know:

1. Basic Linux CLI knowledge;

2. Basic computer networks theory.

How can I start?

First things first, you will need Docker to make the installation, which is an open platform for developing, shipping, and running applications in so-called containers. If you are using a Linux operating system, you can install it using your terminal. In debian based distros (MX Linux, Linux Mint, Ubuntu, etc.) you can use the `sudo apt-get install docker.io` command to conclude the docker package installation.

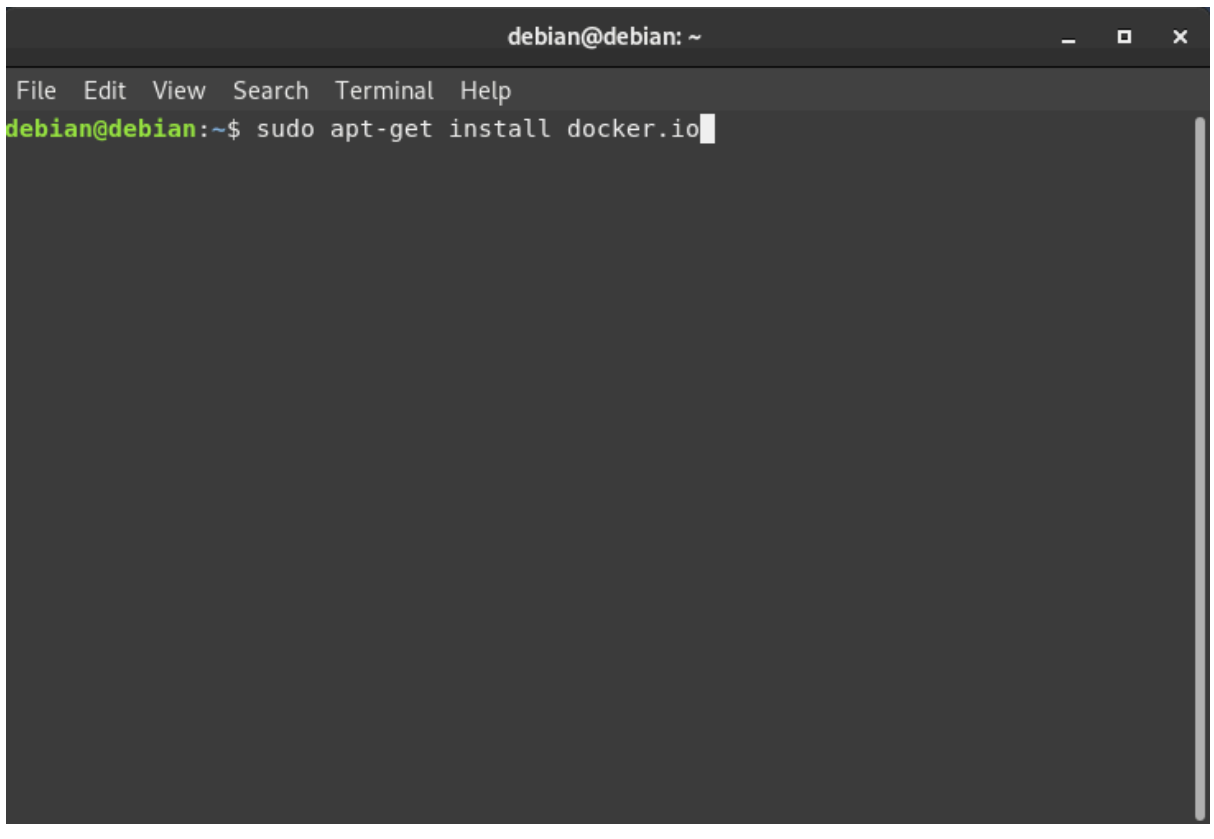


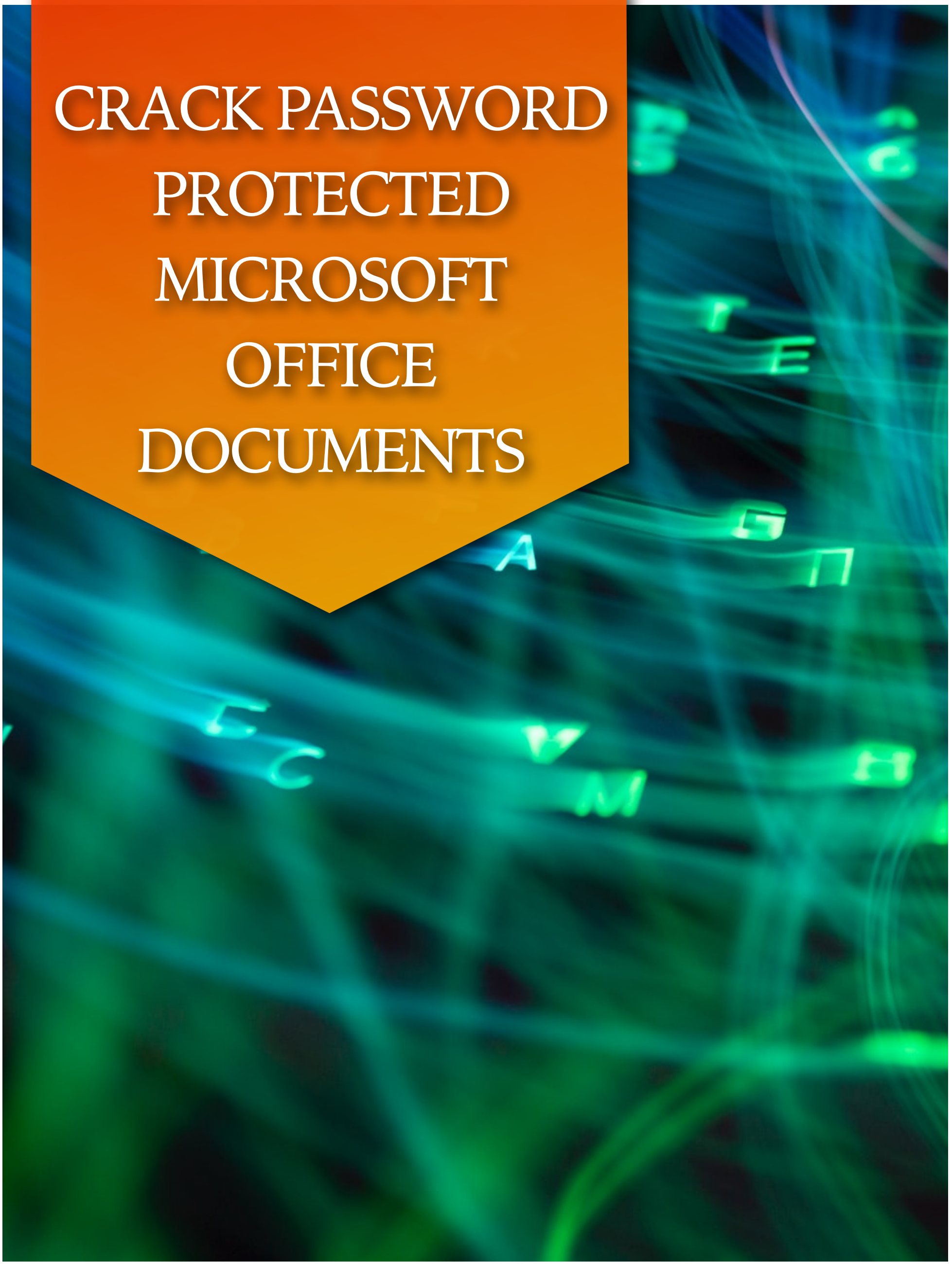
Image 1 - Docker installation command on Debian 10 Buster

For Arch based distros (Manjaro, Endeavour OS, Garuda Linux, etc.) use the `sudo pacman -Sy docker` command instead.

After this, use:

```
sudo docker run -d -p 443:443 --name openvas mikesplain/openvas
```

CRACK PASSWORD PROTECTED MICROSOFT OFFICE DOCUMENTS





DR. AKASHDEEP BHARDWAJ

Dr. Akashdeep Bhardwaj is working as Professor (Cybersecurity & Digital Forensics) with University of Petroleum & Energy Studies (UPES), Dehradun, India. Dr. Akashdeep is an eminent IT Industry & Academic expert with over 26 years of experience in Cybersecurity, Digital Forensics and IT Management Operations. In his current role, Dr. Akashdeep mentors graduate, masters and doctoral students apart from leading projects. Akash has published over 65 research papers, books, chapters and patents. Akash has worked as Technology Leader and head in several multinational organizations.

Introduction

The password protection of Microsoft Office files prevents falsification and ensures data integrity. However, documents with passwords secured from previous versions of Office may be retrieved and hashes extracted. It just takes a few seconds to recover hashes from Microsoft Office files that are password-protected. Although encryption standards have changed throughout the years across various Office programmers, neither can handle some of the instruments utilized in this laboratory experiment.

Tools required:

- Kali Linux
- JohnTheRipper
- Office2John
- Hashcat

Step 1: Install Office2John

```
~# wget https://raw.githubusercontent.com/magnumripper/JohnTheRipper/bleeding-
jumbo/run/office2john.py

Resolving [?] raw.githubusercontent.com (raw.githubusercontent.com)... 192.168.1.1
Connecting [?] raw.githubusercontent.com (raw.githubusercontent.com)| 192.168.1.1|:
443...

HTTP request [?] sent
Awaiting response [?] 200 OK
Length [?] 131691 (130K) [text/plain]
Saving [?] office_john.py

Office_john.py
100% [=====================================================>]
130.66K  --.-KB/s in 019s

2021/09/19 (1.5 MB/s), office_john.py saved [131691/131691]
```

Step 2: Keep all files in the Same Directory

Modify the directory in the same directory in which Office2john is running using Python. This is default but a separate directory can be used as well.