

A surreal, apocalyptic illustration. A massive, bright orange sun dominates the center of the sky. Several brain-like structures, resembling raspberries, float in the sky, with some dripping red liquid. The sky is filled with stylized, layered clouds in shades of orange, red, and teal. In the foreground, two cityscapes are visible on islands, partially submerged in dark blue, turbulent water. The overall style is reminiscent of mid-20th-century pulp magazine art.

**Cloudy
with a Chance of
Malware**

Overview

husky@kali:~\$ whoami

Malware Analysis
Lab, the traditional
way



Move it to the
cloud!



Considerations,
Security, & Guidelines



Tech Demo

Questions / Comments / Thank You!

Matt Kiely / HuskyHacks

huskhacks.mk@gmail.com

@HuskyHacksMK

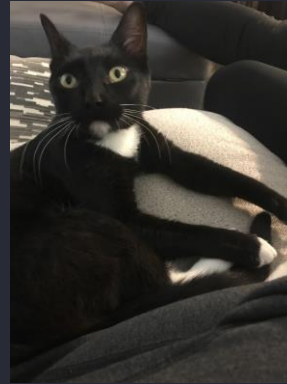
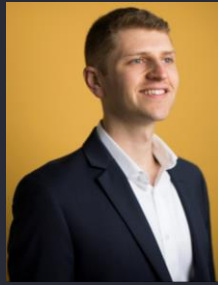
<https://notes.huskyhacks.dev>

Practical Malware Analysis & Triage: <https://bit.ly/tcm-pmat>

husky@kali:~\$ whoami

husky@kali:~\$ whoami

- ▶ Matt Kiely
- ▶ Guy trying to learn stuff everyday
- ▶ Cat dad (Cosmo & Kiki)
- ▶ Appalachian Trail Thru Hiker (class of '23)
- ▶ Red teamer, malware reverse engineer
- ▶ USMC Vet
- ▶ MIT Lincoln Lab
- ▶ Content Author & Instructor
 - ▶ TCM Security
 - ▶ Co-Founder: The Taggart Institute
- ▶ Twitter: @HuskyHacksMK
- ▶ Blog/Notes:
<https://notes.huskyhacks.dev>



Cosmo!



Malware Analysis Lab, the traditional way



-
- Around 38k enrollments across the globe (TCM Security Academy)
 - Teaches the art and science of malware analysis in an approachable way
 - Centers on practical labs, training, and challenges
 - This course requires you to roll up your sleeves and dissect malware in a lab
 - Gives the student the tools of the industry malware analyst (set them up for success!)



PRACTICAL MALWARE ANALYSIS & TRIAGE

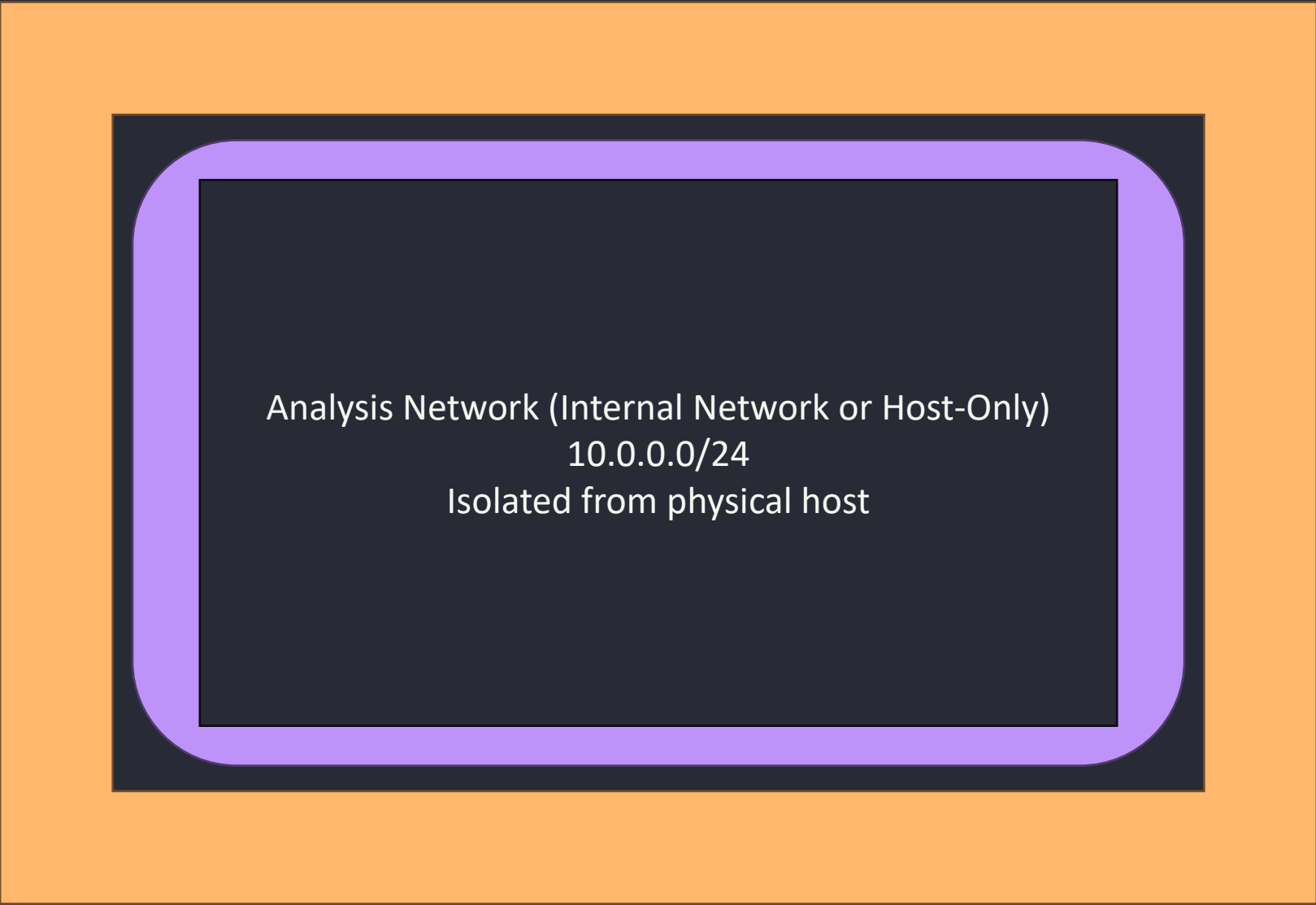


TCM
SECURITY

Student's personal computer

The diagram illustrates the architecture of a Hypervisor (VirtualBox). It features a central light purple rounded rectangle containing the text "Hypervisor (VirtualBox)". This rectangle is enclosed within a black rectangular border, which is itself surrounded by a thick orange rectangular frame. The entire composition is set against a dark blue background.

Hypervisor (VirtualBox)

The diagram consists of three nested rectangular frames. The outermost frame is orange. Inside it is a dark blue frame. Inside the dark blue frame is a light purple frame with rounded corners. The text is centered within the light purple frame.

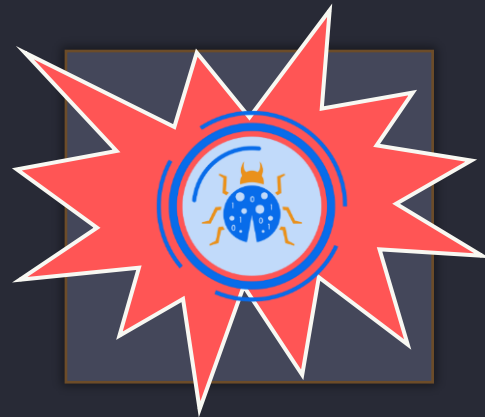
Analysis Network (Internal Network or Host-Only)
10.0.0.0/24
Isolated from physical host

The diagram illustrates a virtual machine (VM) environment. It features a large orange rectangle representing the host. Inside this host is a dark blue rectangle representing the VM environment. Within the VM environment is a light purple rounded rectangle representing the internal network. Inside this network are two gray squares representing the VMs: 'FLARE-VM (Windows 10 or Server)' on the left and 'REMnux (Special Ubuntu Distro)' on the right. Below these two VMs, centered within the purple rounded rectangle, is the text '10.0.0.0/24', indicating the IP address range for this internal network.

FLARE-VM
(Windows 10 or
Server)

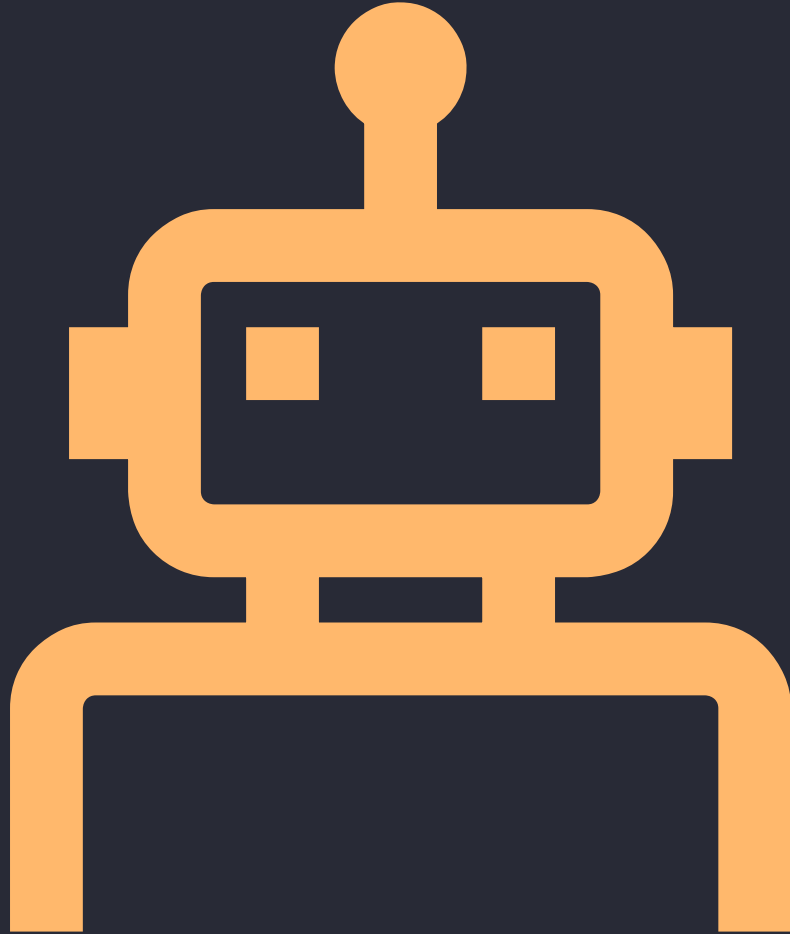
REMnux (Special
Ubuntu Distro)

10.0.0.0/24



REMnux (Special
Ubuntu Distro)

10.0.0.0/24



Malware Analysis lab, the traditional way

- Local virtual machines (VMWare, VirtualBox, etc)
- Doesn't scale well
- Differences in student setups means harder to troubleshoot
- Takes time to set up (FLARE-VM installation, primarily)

Student (Analyst) Perspective - Common Challenges

- Inability to run the specified hypervisor (Mac m1/m2 chips)
- Variability in personal computer specs, VM performance
- **Risk!** “Is this really safe to do on my personal computer?”
- Can we benefit from moving to the cloud? **Yes.**



Move it to the
cloud!




```
graph LR; A((PMAT student versed in AWS (@Flekyy90) pitches cloud lab idea. Brilliant!)) --> B((Stack: AWS, pre-built AMIs (FLARE-VM, REMnux), Apache Guacamole, Terraform)); B --> C((Security: network egress/ingress rules to isolate lab when detonating malware)); C --> D((Result: rapid-deployable cloud malware lab that stands up in a few minutes.))
```

PMAT student
versed in AWS
(@Flekyy90)
pitches cloud lab
idea. Brilliant!

Stack: AWS, pre-
built AMIs
(FLARE-VM,
REMnux), Apache
Guacamole,
Terraform

Security: network
egress/ingress
rules to isolate lab
when detonating
malware

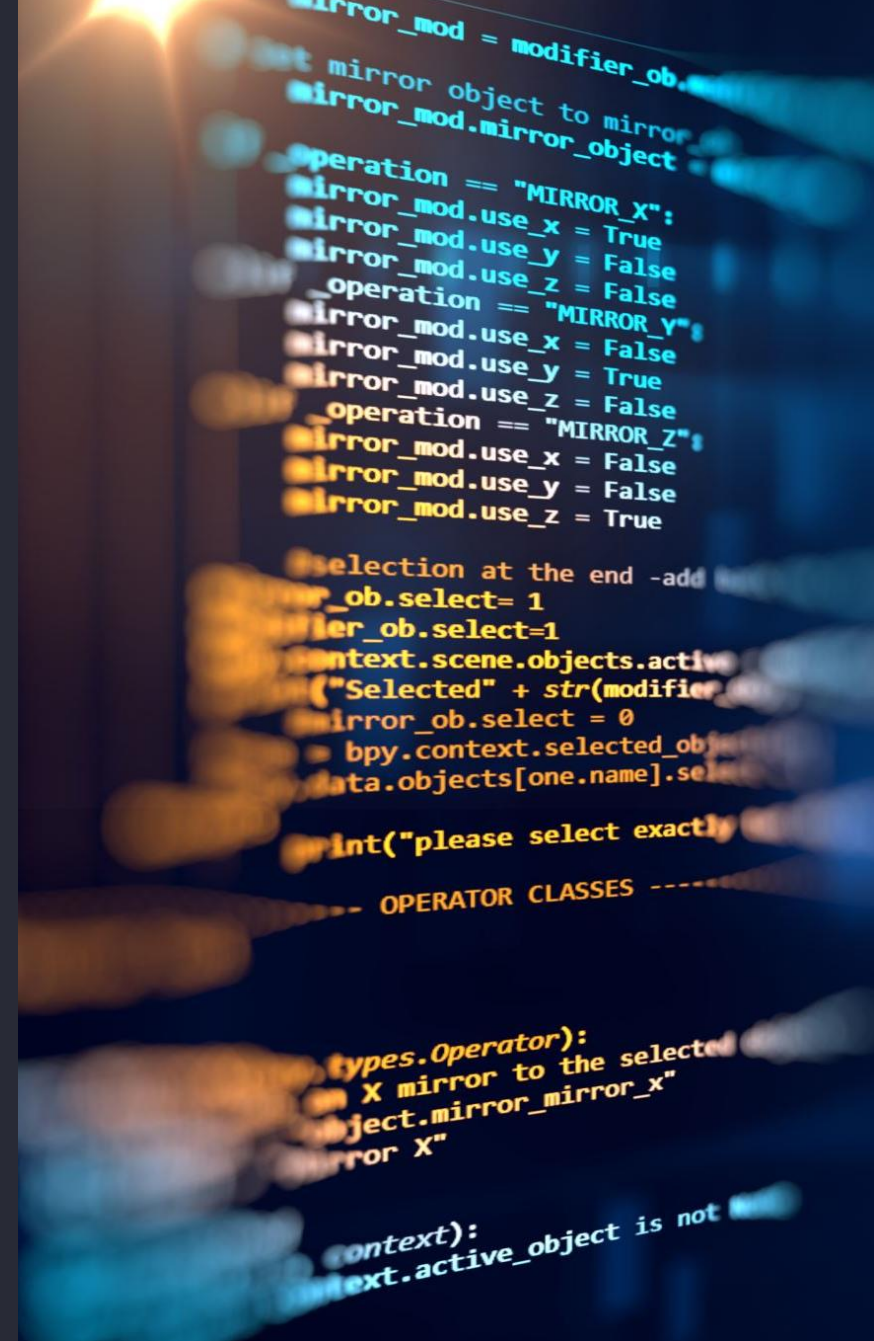
Result: rapid-
deployable cloud
malware lab that
stands up in a few
minutes.



Considerations, Security, & Guidelines

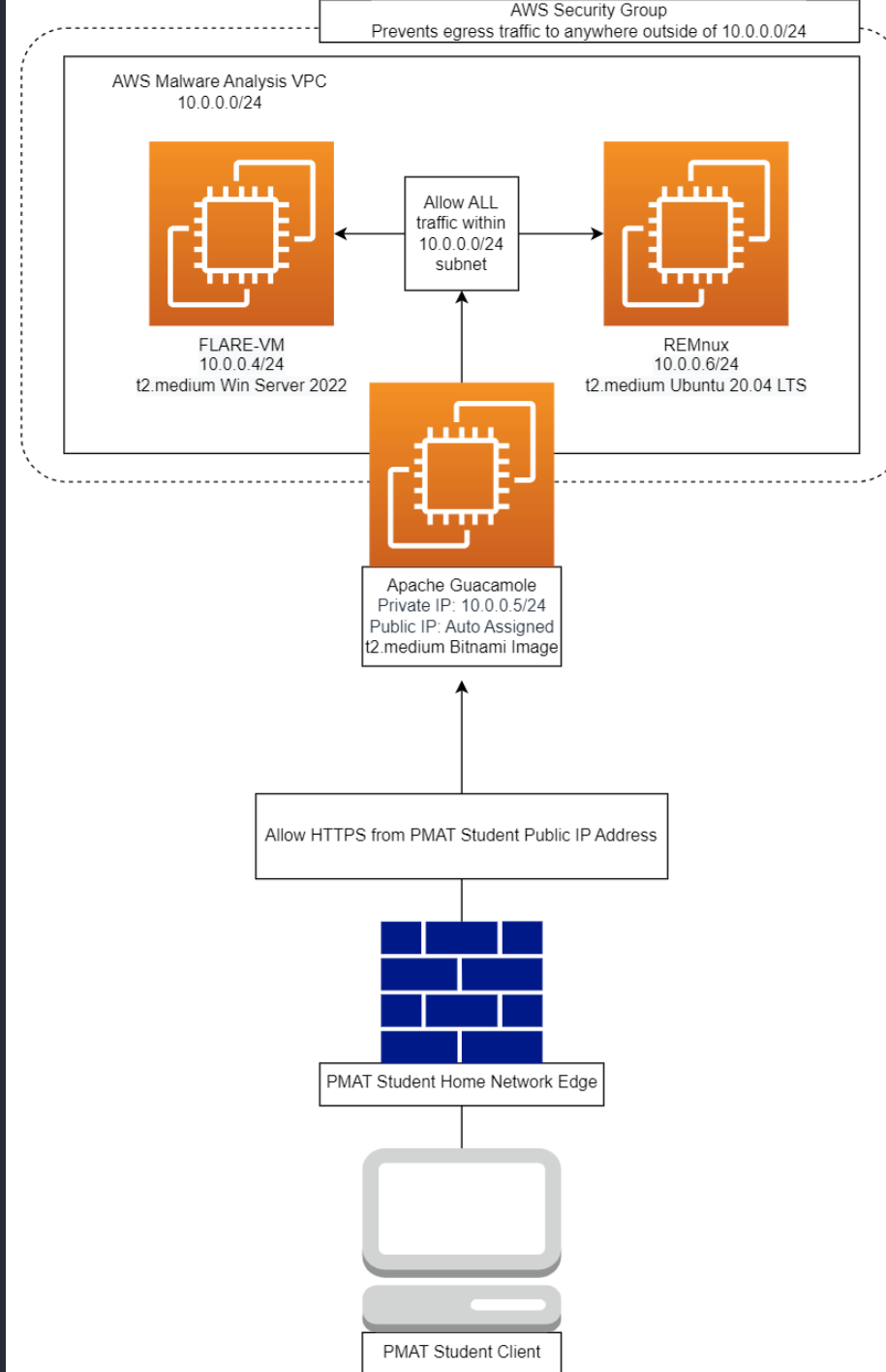
Considerations, Security, & Guidelines

- AWS requires you to submit for permission to perform malware analysis (the Simulated Events form)
 - <https://console.aws.amazon.com/support/contacts#/simulated-events>
- Some assembly required
 - ...as with all cloud provisioning, there's some setup (IAM, configuring aws-cli, getting Terraform to behave)
- Low cost but not without cost
 - As a point of reference, estimated \$4 for instance time to complete the course
 - I foot the bill for the pre-built AMIs 😊



General Requirements

- This will be carried out in a secure VPC.
- The VPC and instances will have inbound traffic restricted to a set of IP addresses owned by the customer.
- The instances involved will not have public IP addresses.
- The instances will not be allowed to send any packets to the internet (to include via proxies).
- DNS should be disabled in the VPC to prevent malware looking up command and control domains.
- Malware should be detonated in a sandbox.
- Systems involved should be fully patched and hardened in accordance to security best practices.
- System monitoring and logging should be in place and reviewed.
- Simulation services, such as INetSim are allowed but must be run within the same VPC as the malware.
- Secure S3 bucket and have encryption turned on.



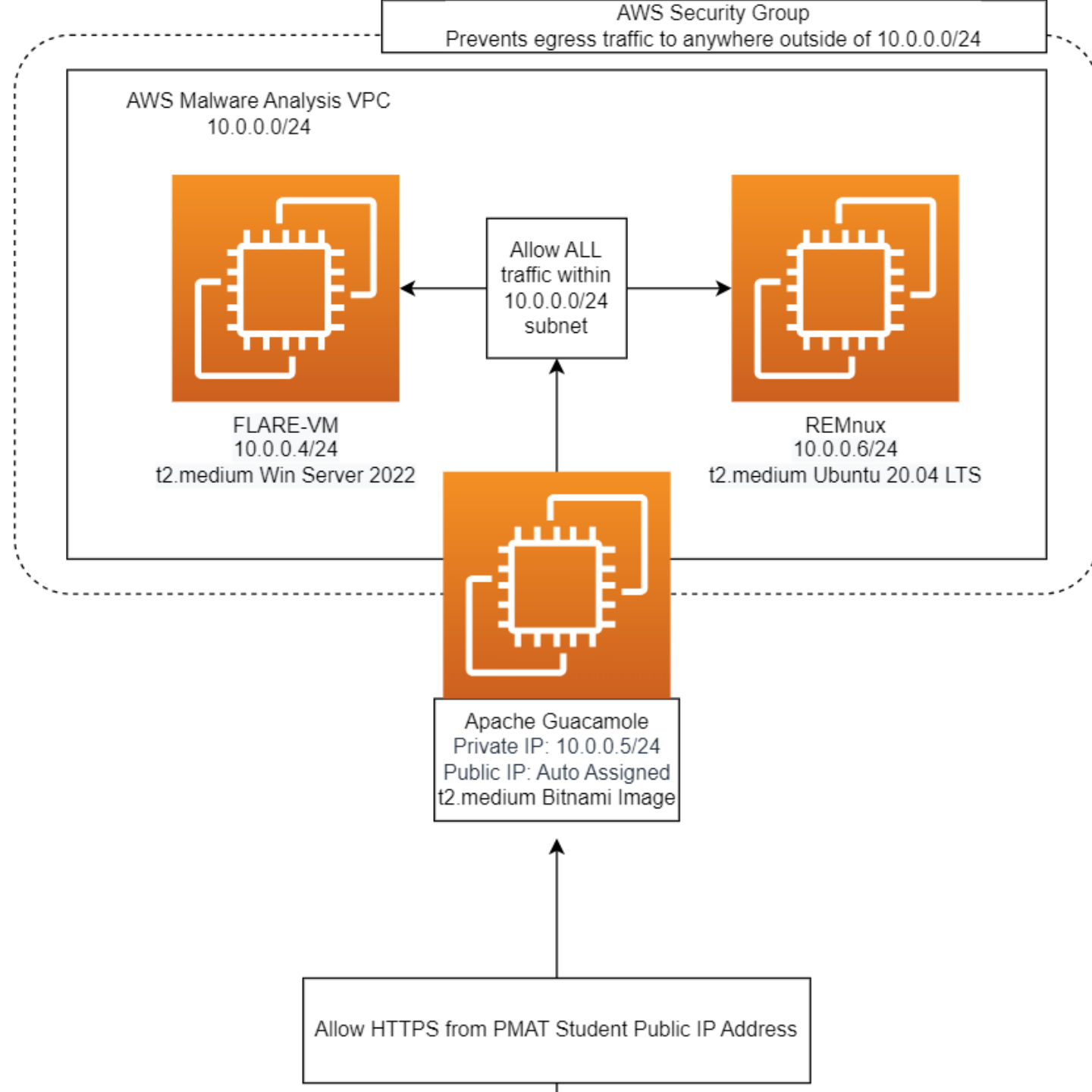
Allow HTTPS from PMAT Student Public IP Address

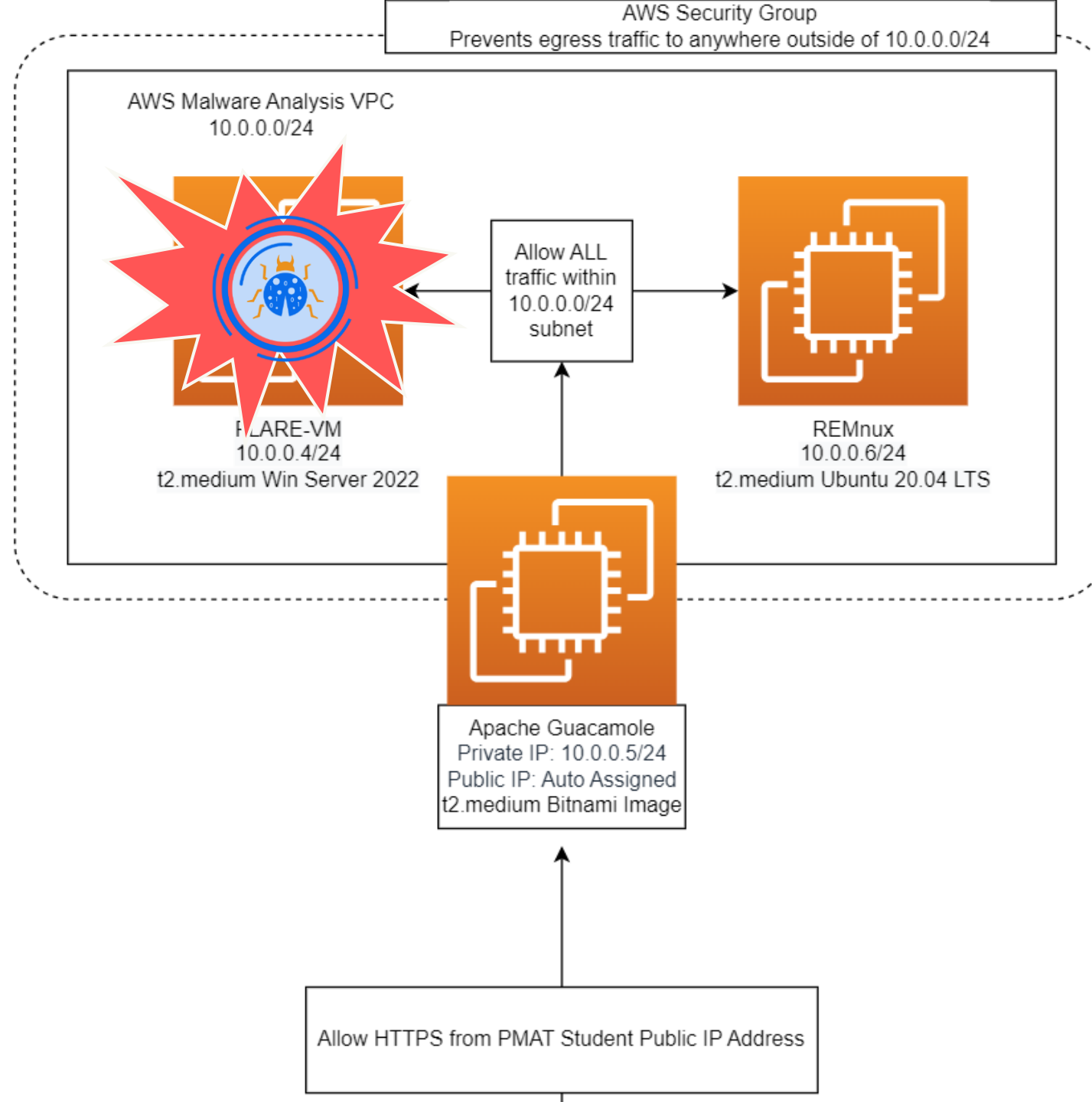


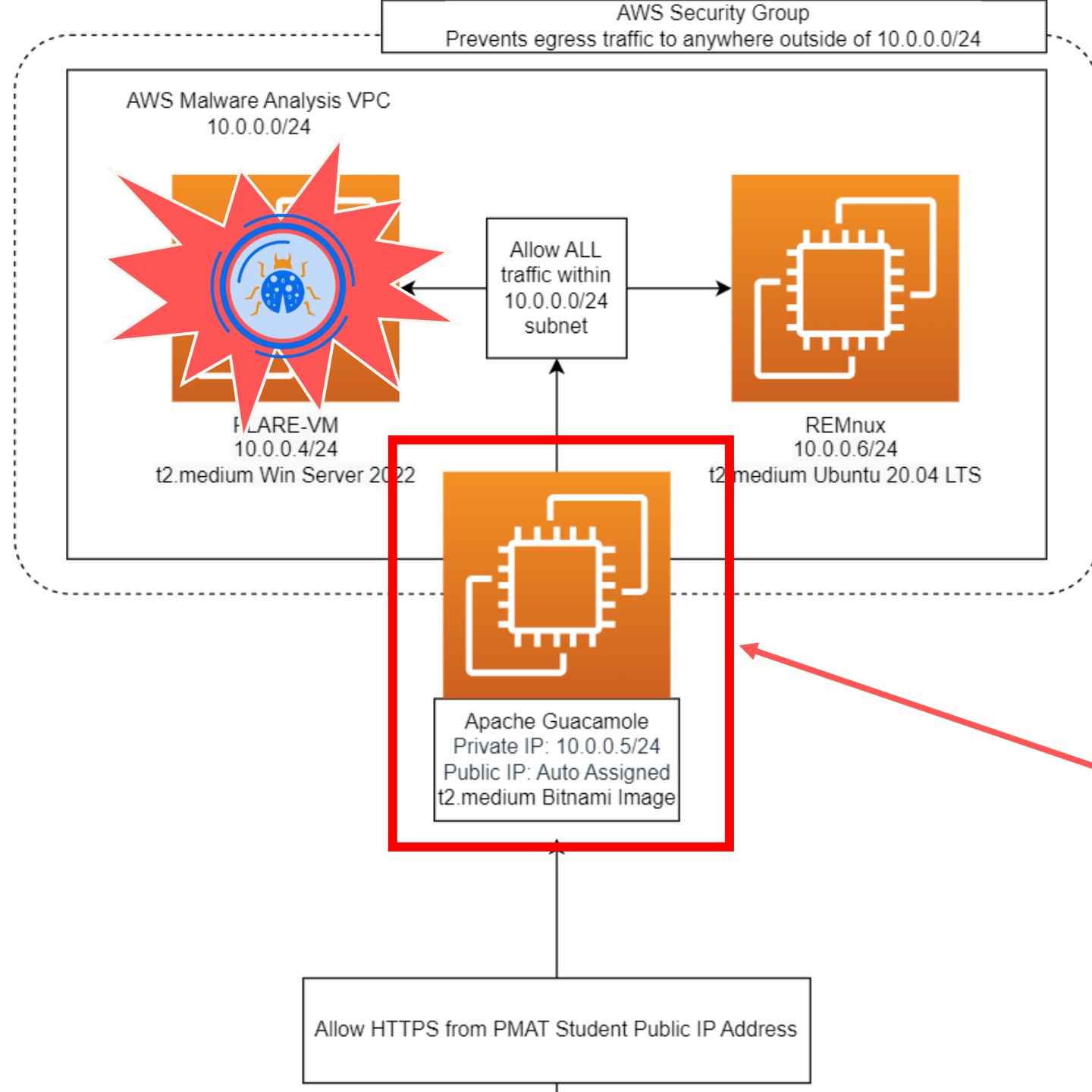
PMAT Student Home Network Edge



PMAT Student Client







Our vantage point
into the lab!

Pause! Hit go on the
tech demo lab!

Security by Architecture, Architecture as Code

- Terraform makes the configuration identical, each time, every time
 - ... with a few exceptions
 - Student's home IP is discerned at runtime by Terraform and used to set up the security group rules
- VPC, subnet, ingress/egress are all handled by Terraform
- Provisioned without internet access by default, can manually add internet access security group to download malware/software

```
169 # Create Security Group for Guacamole
170 resource "aws_security_group" "security_group_guacamole" {
171     count          = var.enable_guacamole ? 1 : 0
172     name           = "security_group_guacamole"
173     description    = "Allow HTTPS from the Internet"
174     vpc_id         = aws_vpc.lab_vpc.id
175
176     ingress {
177         description      = "Allow HTTPS inbound traffic"
178         from_port        = 443
179         to_port          = 443
180         protocol         = "tcp"
181         cidr_blocks      = ["${chomp(data.http.myip.response_body)}/32"]
182     }
183
184     egress {
185         from_port        = 0
186         to_port          = 0
187         protocol         = "-1"
188         cidr_blocks      = ["0.0.0.0/0"]
189         ipv6_cidr_blocks = [ "::/0" ]
190     }
191
192     tags = {
193         Name = "${var.environment}-guacamole"
194     }
195 }
```

- Let FLARE-VM talk to everything in the VPC...
- ... but don't let it talk to anything outside the VPC!

```
85 # Create Security groups FlareVM - no internet
86 resource "aws_security_group" "security_group_flarevm_no_internet" {
87     count          = var.enable_guacamole ? 1 : 0
88     name           = "security_group_flarevm_no_internet"
89     description    = "Allow inbound from local subnet"
90     vpc_id         = aws_vpc.lab_vpc.id
91
92     ingress {
93         description = "Allow inbound traffic from local subnet"
94         from_port   = 0
95         to_port     = 0
96         protocol    = "-1"
97         cidr_blocks = ["10.0.0.0/24"]
98     }
99
100    egress {
101        description = "Allow outbound to local subnet"
102        from_port   = 0
103        to_port     = 0
104        protocol    = "-1"
105        cidr_blocks = ["10.0.0.0/24"]
106    }
107
108    tags = {
109        Name = "${var.environment}-flarevm-no-internet"
110    }
111 }
```




Tech Demo

Questions / Comments / Thank You!

Matt Kiely / HuskyHacks

huskhacks.mk@gmail.com

@HuskyHacksMK

<https://notes.huskyhacks.dev>

Practical Malware Analysis & Triage: <https://bit.ly/tcm-pmat>

References

- [Michael Mattes: Malware analysis with AWS \(2022\)](#)
- <https://github.com/adanalvarez/AWS-malware-lab>