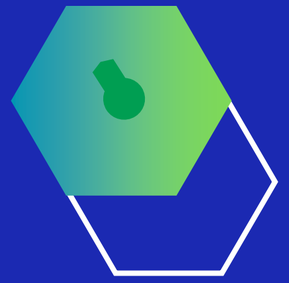
Enterprise Defence

Hardening

Threat Vectors



Networks

- While networks don’t make the first endpoint, it does make a second endpoint, especially once someone has received malware.
- Of course, this goes beyond the usual router and firewall hardening
- We will require to introduce SOC teams, dynamic network analysis, cloud hardening and other measures.

Web Applications

- This is the second large threat vector: Your applications being hacked, i.e. misused to break the CIA triad.
- Of course, this goes beyond OWASP Top 10 and avoiding auth.js
- We will require to introduce secure coding principles, supply chain mitigations among other measures.

Other Phishing

- This includes unorthodox but common methods like:
- Hardware Jacking
- QR Spoofing
- Voice Phishing
- Insider Threats
- Bad USB

Network Principles

Defence in Depth

Layering Defences

SOC Tactics

Monitoring and
Responding to Threats

01

Defence in Depth

Layers

Access Control

Preventing bad traffic from entering networks

SOC Control

Detecting and removing bad traffic on networks

Isolation

Hiding key assets and data from network

Access Layers

Second Touch

Paid Router with Built-In Antivirus with additional Access Control Lists

Third Touch

Secure Switches:

1. OPNSense Firewalls behind major switches
2. Enable max security for Switches

First Touch

Paid Stateful Firewalls: Harden with additional blacklists, e.g. DNS Control, Spamhaus, Firehol

Centralizing

Set up Ansible to automate updates of additional lists and firmware

SOC Layers

SOAR

Add a personal regression model for threat severity and false positive likelihood; trained on public datasets

EDR

Helps to provide a single platform to deal with all SOAR alerts. Choose on basis of ease of use; simple EDRs can be used by network staff

SIEM

Ensure it focuses on high-value IoCs: Known bad files, Failed/Untimely MFA registrations, temp file executions etc.

Centralizing

Get developers to make you intranet hosted middleware to work with all the above 3 using their REST APIs

Isolation Layers

App Deployment

The layer that works with the critical layer is on the cloud; hardened with IAM and Cloud Security Policies (Max)

Critical Layer

Stay on-premises on a separate VLAN, and tight Access Control Lists with Next Gen Firewall between VLANs.

Work Machine

Keep development and similar machines on a microsegment, limiting application-layer access

Centralizing

Avoid Centralization, it is a STRICT NO-NO

02

SOC Tactics

Threat Intelligence

Set up Dashboard

Allows you to see threats relevant to you quickly, use Grafana/Metabase

Data Sources

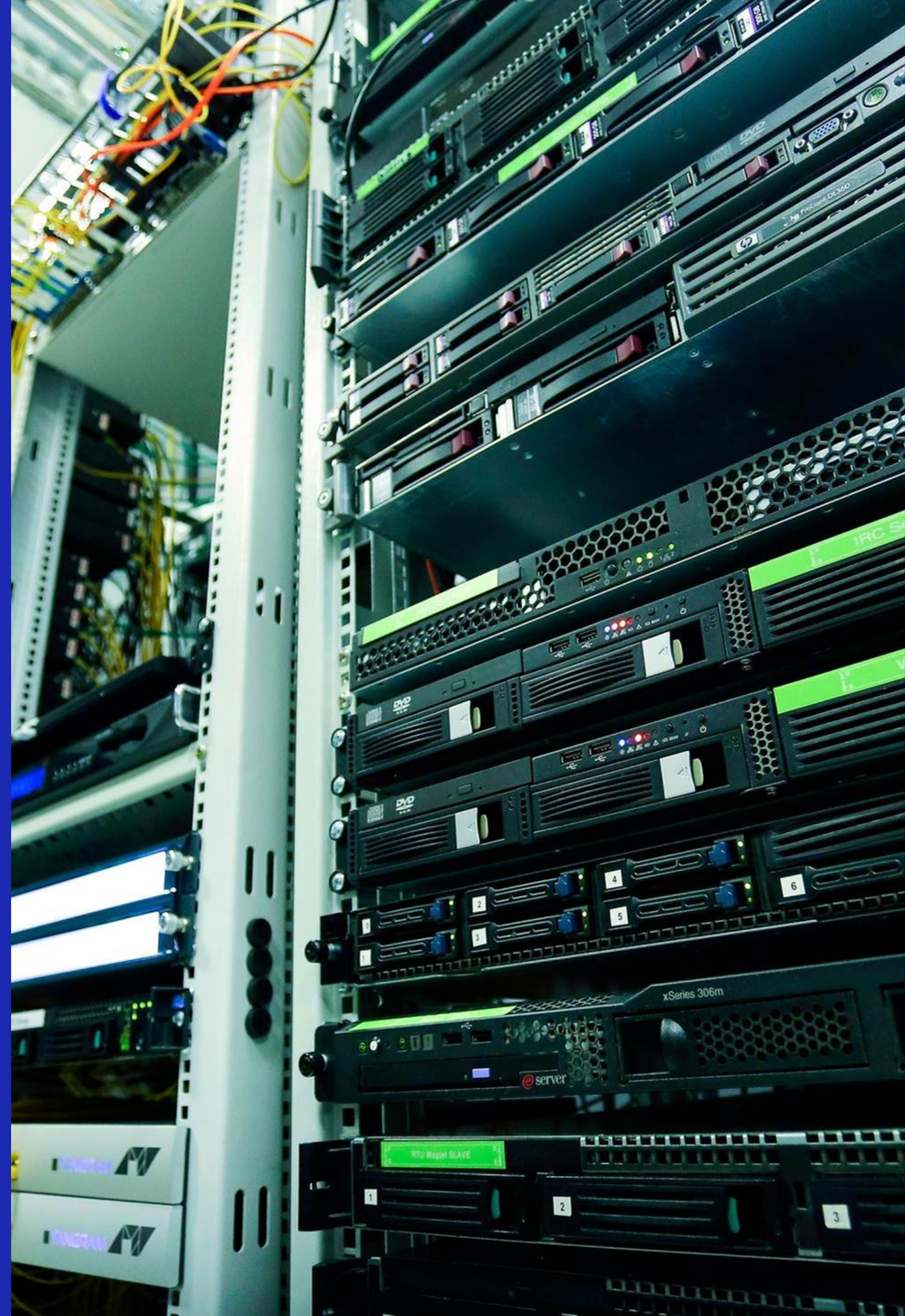
Get all you can: Shodan, Censys, HIBP, Abuse.ch, Spycloud, Greynoise

Tech Stack

Python scripts to pull data from APIs, SQL to prepare the data and Cron to push it to Grafana

Important Widgets

Publicly exposed assets, IOCs matching, Leaked Credentials, Mentions of company in bad forums



Studying Attacks

Isolate via DNS

Using DNS Sinkholes, re-route the traffic into honeypots to study attacker behaviour

Dummy Data

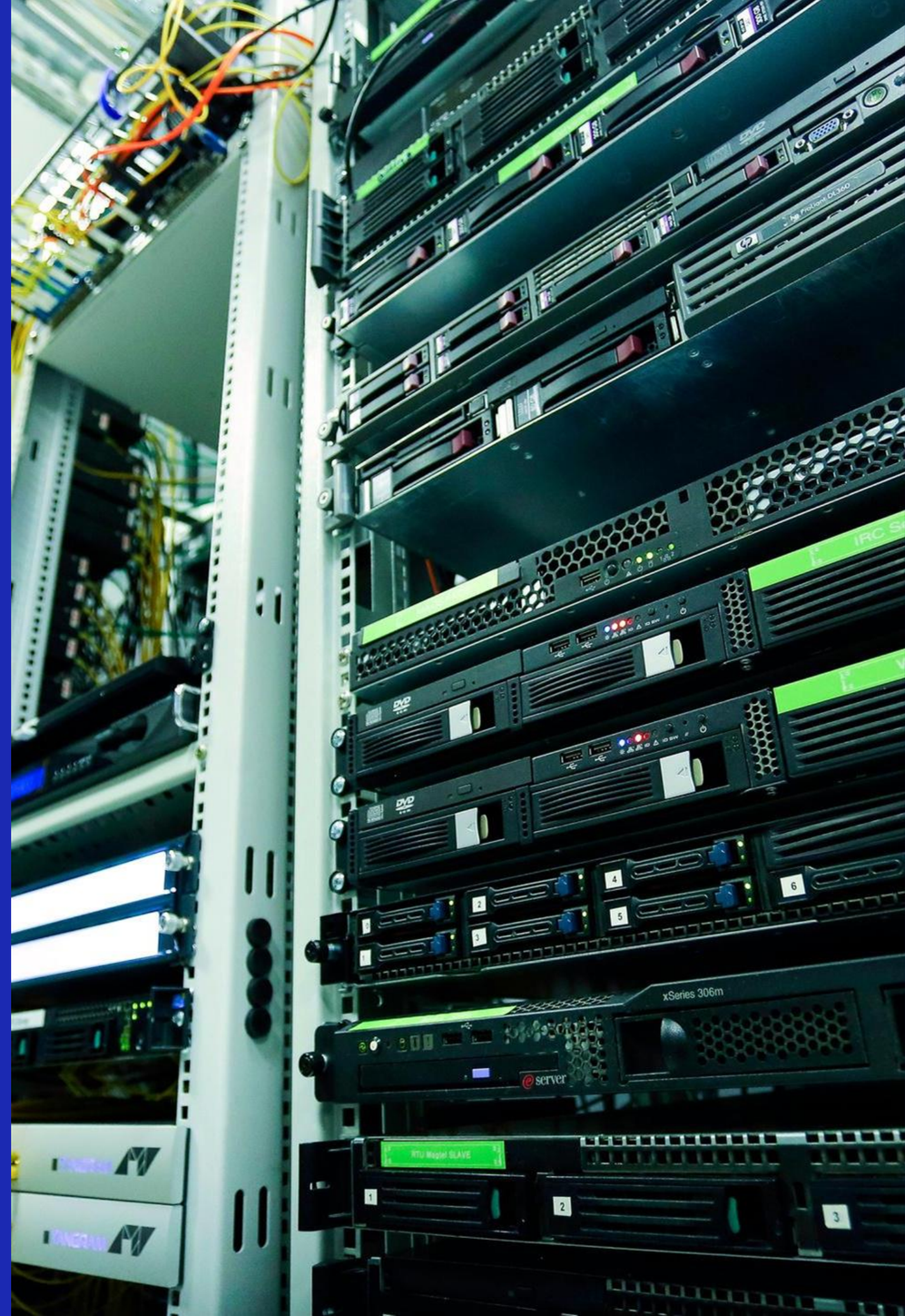
Ensure the honeypot is full of enticing files filled with interesting fake data

Honeypot Services

Simulate internal applications such as API endpoints logging inputs, Admin Panels etc.

Command Analysis

Ensuring logging cannot be disabled on honeypot ensures you can easily analyse attack later



Profiling IoCs

Identity + Legitimacy

Take username/IP etc. with any one indicator of legitimacy level “E.g. CEO logged in from Nigeria at 12AM”

Process Tree

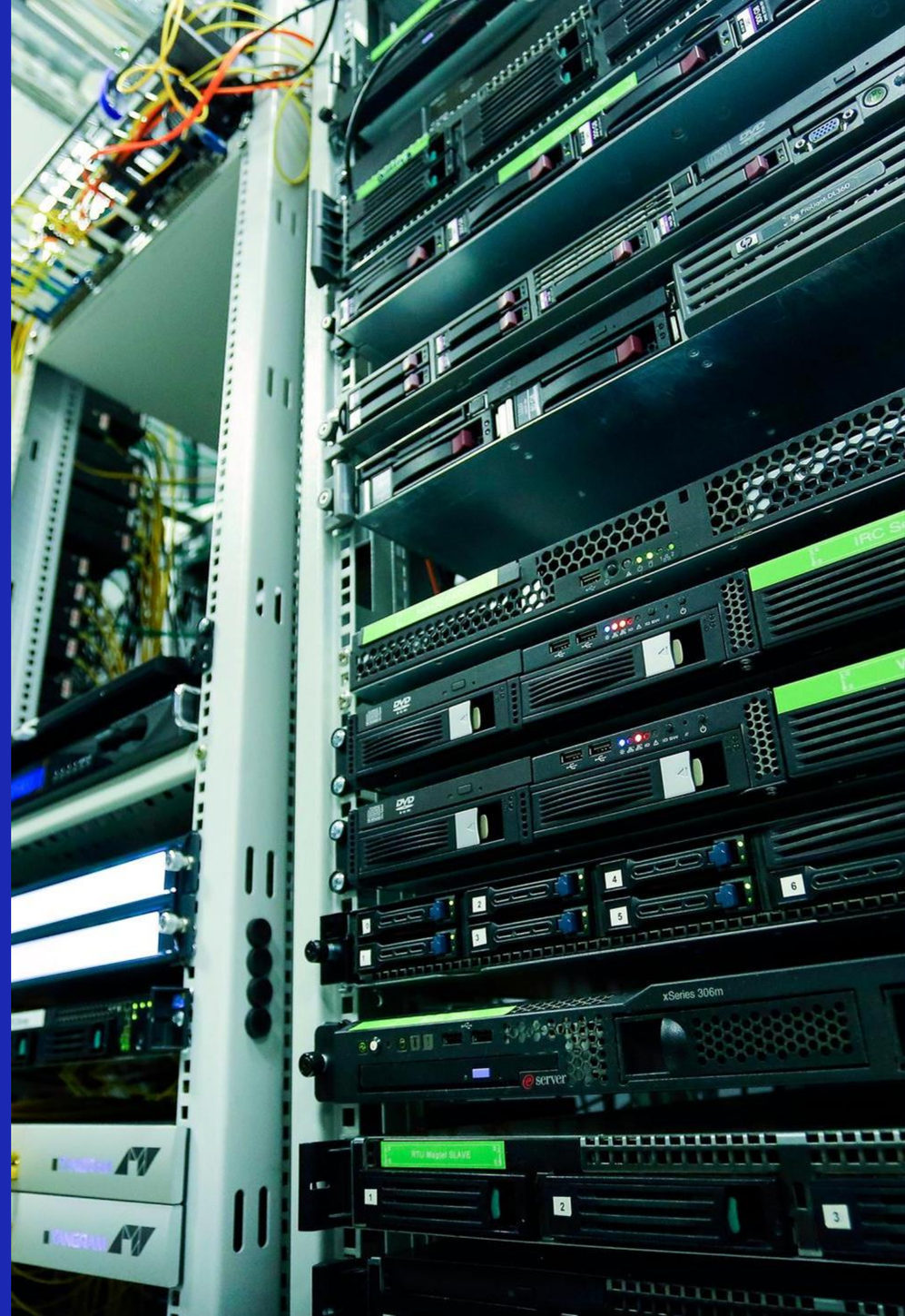
Denote process relationships with arguments E.g. “winword spawned powershell with –nop –enc”

Network Behaviour

Take destination IP/DNS, port and repetition e.g. “Outbound HTTP to 185.204.168.196 every 10 seconds

Data Touched

Any data interacted with such as File paths/Config changes “e.g. Macro added to Invoices2024.xlsx”



Responding afterwards

Stop the Attack!

Web Applications

AppSec Principles

Minimal Touch

Continually validate, isolate and sanitize inputs at every layer

Safe Failing

Design all failure responses with intention to avoid hacks

Minimal Touch

Treat Raw Data Like Lava: Only let it touch specially cooled pipes!

Validate THEN Convert into Immutable Object

Isolate Input Processing with Sandboxes (only using Parameterized Input)

Log all User Input and Log Flagged Inputs Especially

Safe Failing

Treat Raw Data Like Lava: React immediately if it touches you!

Throw flagged input immediately with an error

Ensure error handling using sandboxes output

Set Placeholder Output for Encoding Failures

03

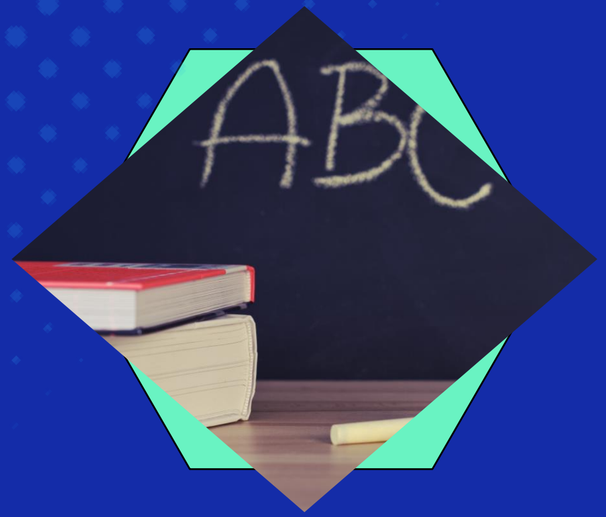
Phishing

There's no conceivable system that can stop 1 person in 100 opening a phishing email and that can be all it takes

Advanced Phishing

Is not Nigerian Prince scams, it is Spear Phishing, Malicious USBs left around, replacing QR codes on nearby restaurants, Dating Scam to install malware on employee mobiles; and everything else imaginable.

Solution



Quarterly Training

Frequent Employee Training **is crucial**

Physical Access

Strong Physical Access restrictions **help greatly**

Separate Network

If possible, set up a separate network for mobile devices