Fight the Fight

Orchestrating and Automating Your Incident Response Process

March 2021

BSIDES

@PeterMorin123

@petermorin123

peter@petermorin.com

https://www.petermorin.com

Peter Morin

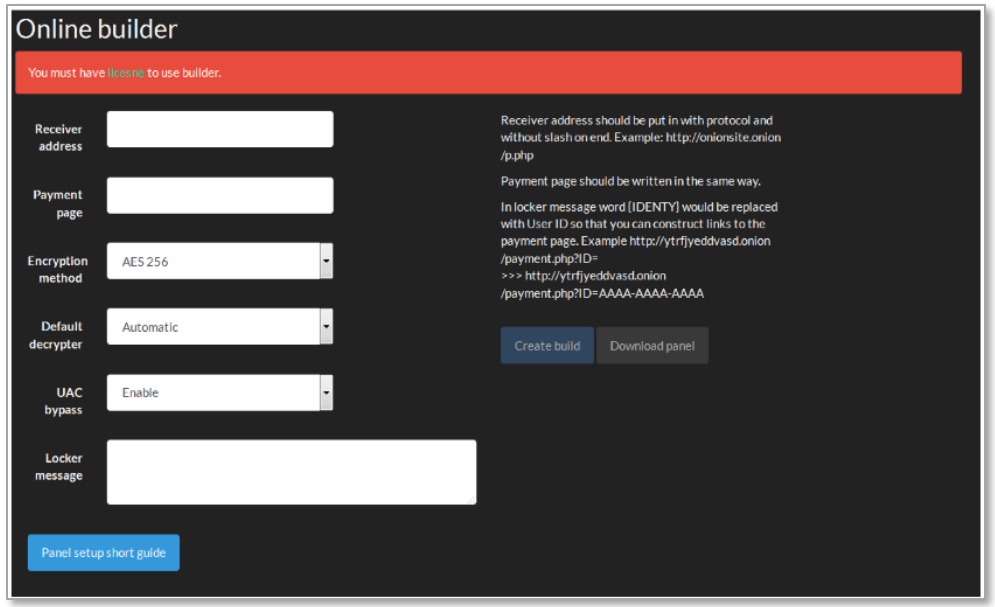
National Cybersecurity Practice Leader
Grant Thornton

- Based out of Halifax, Nova Scotia, Canada
- Over 25 years of experience cybersecurity
- Specialize in security of critical infrastructure, incident response, threat hunting, etc.
- Worked in the past for the various military and government agencies as well as numerous public utilities
- Spoken at events run by Blackhat, FBI, DHS, ISACA, FIRST, US DoD as well as numerous colleges and universities.
- CISSP, CISA, CRISC, CGEIT, GCFA

I am Not a Lawyer | Disclaimer

- The views and opinions expressed in this presentation are mine and do not reflect in any way those of my employer.
- I do not specifically endorse one product over another mentioned in this presentation.
- I do not recommend installing a product or configuring a system without proper design, consultation (with your vendors) and testing.

State of Incident Response | Custom Malware



Increase in eCrime groups, the dark web and the rise of ransomware as a service

- Custom malware is now being used in 50 percent of the attacks demonstrating the scale of the dark web
- Malware and malware services can be purchased to empower traditional criminals, spies and terrorists, many of whom do not have the sophisticated resources to execute these attacks.
- If you have enough money, you can purchase access to an impacted organization without needing much hacking skill

The combination of initial access brokers and ransomware as a service has really lowered the bar of entry into this space for cybercriminals.

State of Incident Response | Island Hopping

Island Hopping putting small businesses at risk of sophisticated attacks.

- Trying to compromise smaller organizations in order to go after their larger partners in the supply chain.
- 55 percent of cyberattacks target the victim's digital infrastructure for the purpose of island hopping.
- Weak e-mail security, identity management (MFA), monitoring controls and endpoint protection.

Breach of Target's point of sale system in 2014 resulting in the theft of payment information from 40M customers and costing 300M dollars in damage, caused by Fazio Mechanical.

State of Incident Response | Increasingly Permeable Perimeter

Attackers Exploiting Trust

- The pandemic has meant a rapid shift to remote work and has expanded the corporate perimeters into employees' homes
- Borders that had been well defined are nw porous
- Companies have been forced to move rapidly to the cloud to support workforce
- Shadow IT bring new cloud applications to the enterprise unknown to the security defenders

This has created a dangerous new threat landscape almost overnight.

State of Incident Response | Counter IR

Incidents of counter IR techniques are at an all-time high, occurring in 82 percent of IR engagements

- Disabling anti-malware, deleting logs, timestamp manipulation sandbox evasion, packing, obfuscation, etc.
- The UK's National Cyber Security Centre reported that one organization paid $9 million for a ransomware decryption key allowing them to recover their files.
- They did not identify the root cause of the attack - same attacker hit them again, using the same mechanism to re-deploy its ransomware.

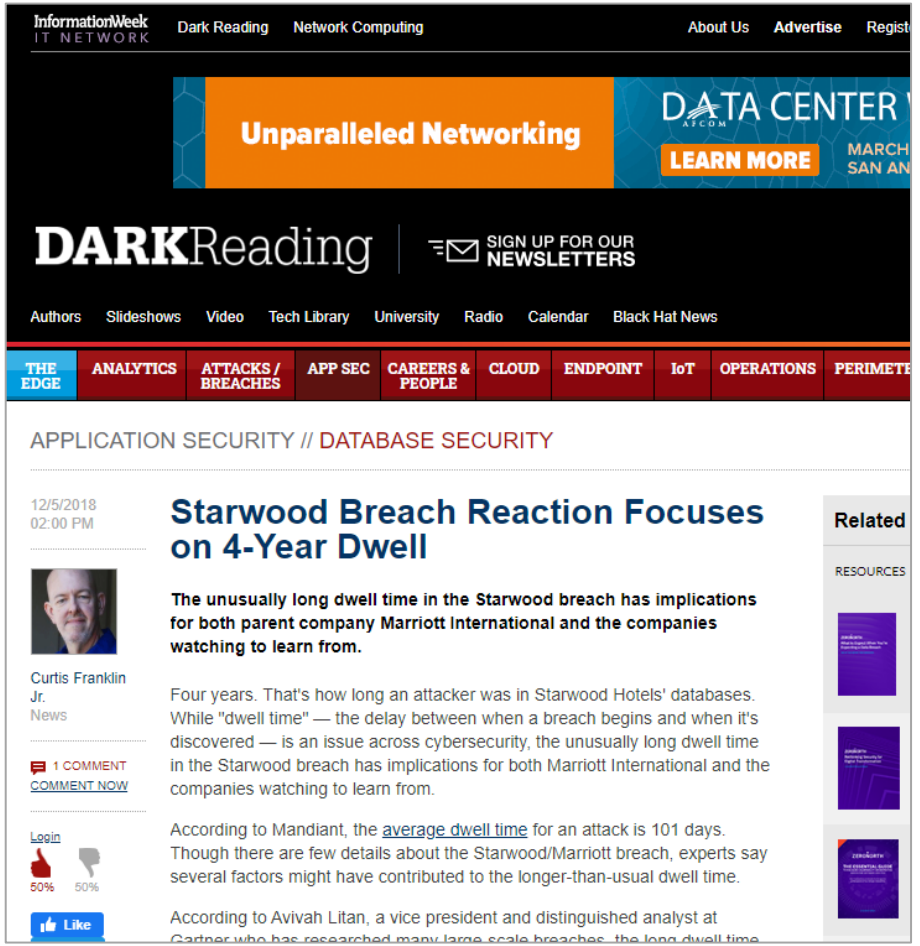
"The victim felt they had no other option but to pay the ransom again," said the UK NCSC.

State of Incident Response | Well Known Organizations Breached

@PeterMorin123

State of Incident Response | "Dwell" Time

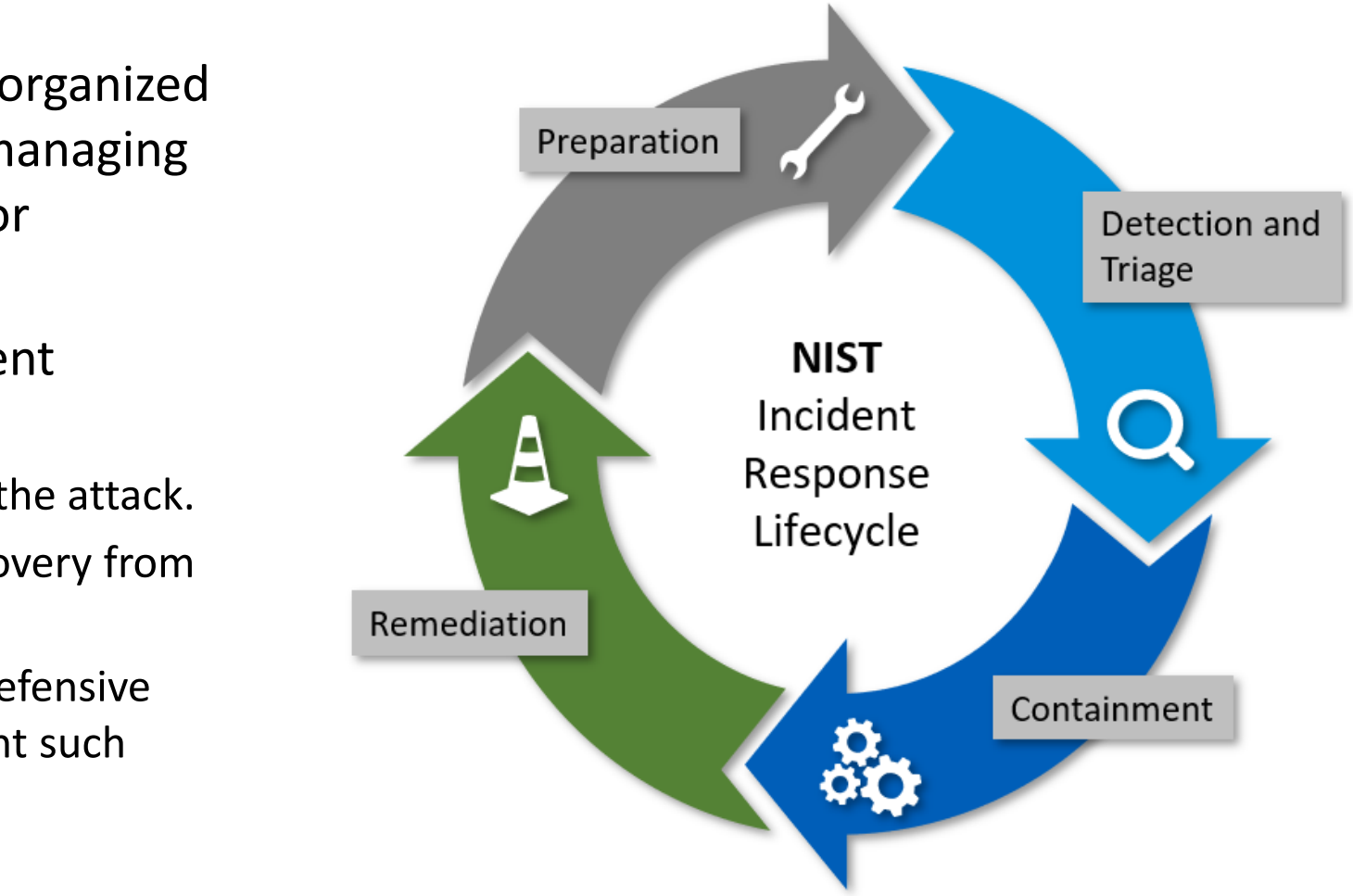
- Time from intrusion to containment
  - Dwell time is down, but still high
  - 56% of breaches took months or more to discover
  - The average threat can lurk undetected for 100+ days
  - **Marriott suffered a 4 year dwell time**



Source: 2019 Verizon DBIR

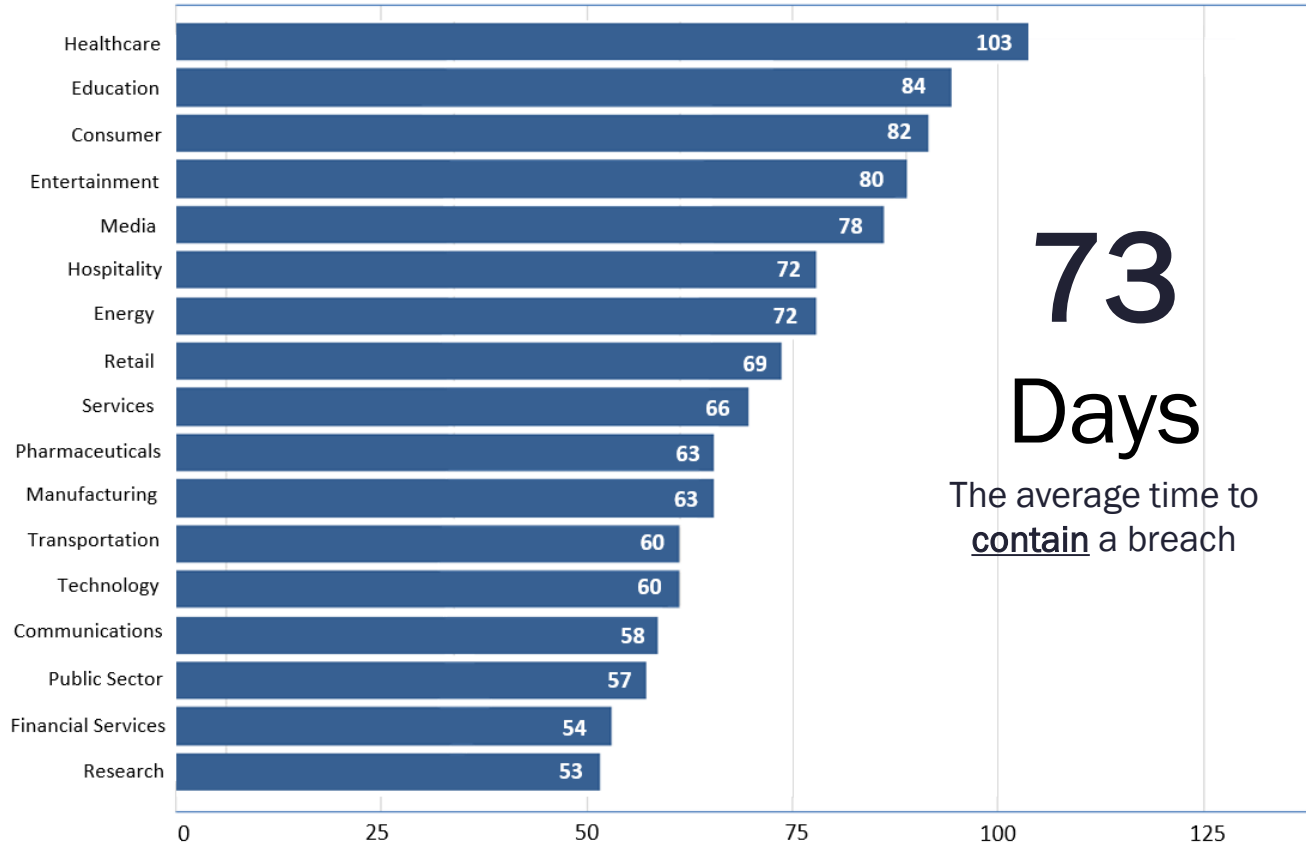
Incident Response Lifecycle | 4 Step Process

- Incident response (IR) is an organized process of addressing and managing the results of an incident (for example, a cyber attack).
- The main goals of the incident response are:
  - To minimize the damage of the attack.
  - To minimize the time of recovery from the attack.
  - To create instructions and defensive measures that would prevent such attacks in the future.



Incident Response | Focus on Containment

- ***Detection is obviously important, but…this is where the "work" begins…***
- Identify the compromised computers and fully understand the scope of the breach and its affected assets and **stop the bleeding**.
- Also **reconfigure** the organization's network to ensure that the existing business **processes would continue running** without the compromised assets.
- This phase can **kill a security team!**



Source: 2020 Ponemon Cost of Data Breach Report

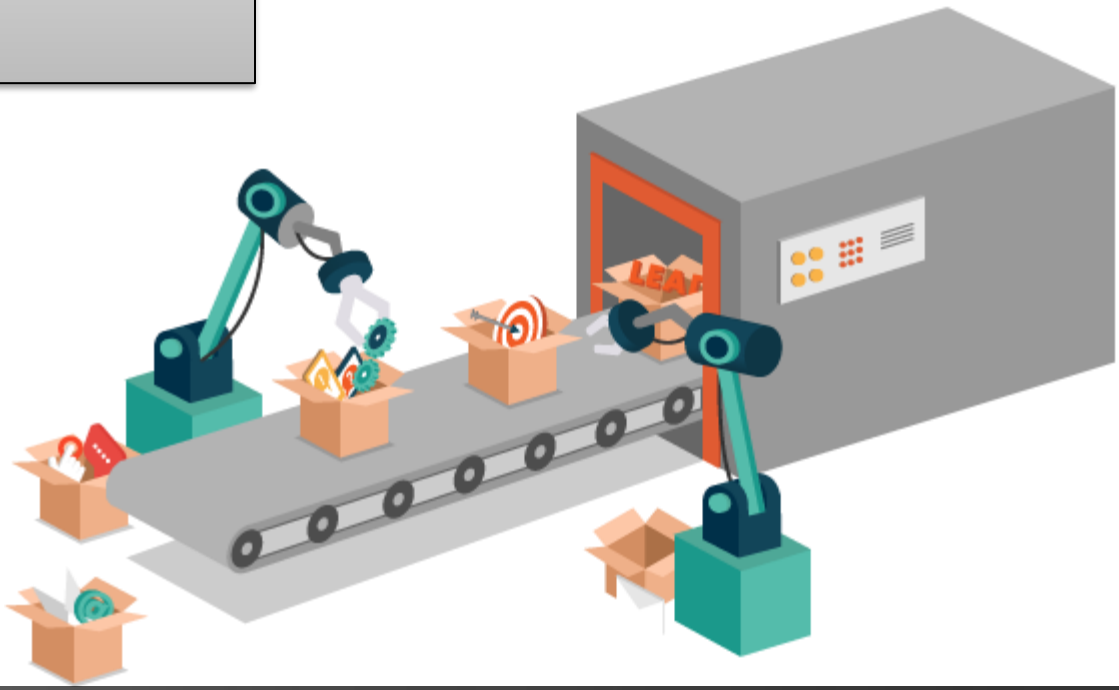
Companies that contain a breach in **less than 30 days save more than $1 million** in comparison to those who take longer.

Source: 2020 Ponemon Cost of Data Breach Report

Incident Response| Focus on Containment

- If one of the servers in the organization's network is compromised by an attacker, the security team must isolate this server from the network.
- The security team must also adjust routing policies to distribute this server's load to other servers
- Tools to enable front-line analysts to react quickly – **enable your tier 1 folks!**
- A lack of solid containment processes could lead with a threat still present and spreading in the environment (i.e. ransomware).

Automation decreases the average response time.
Automating mundane and time-consuming security tasks allows you to allocates your IT and security team's time to higher-level security duties that allow them to take a deeper look into potential threats.



Incident Response| Introducing SOAR

Security **O**rchestration

Automation

Response

Incident Response| What is SOAR?

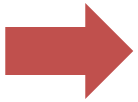
Security Orchestration

- Integrating disparate technologies and connecting security tools (security-specific and non-security specific)
- Make them capable of working together and improving incident response

- What does the threat intelligence data indicate?
- Were similar emails received by any other system?
- What IP address did it come from?

Automation

- Machine-driven execution of actions on IT systems and security tools as a part of incident response.
- These tasks were previously performed by humans.



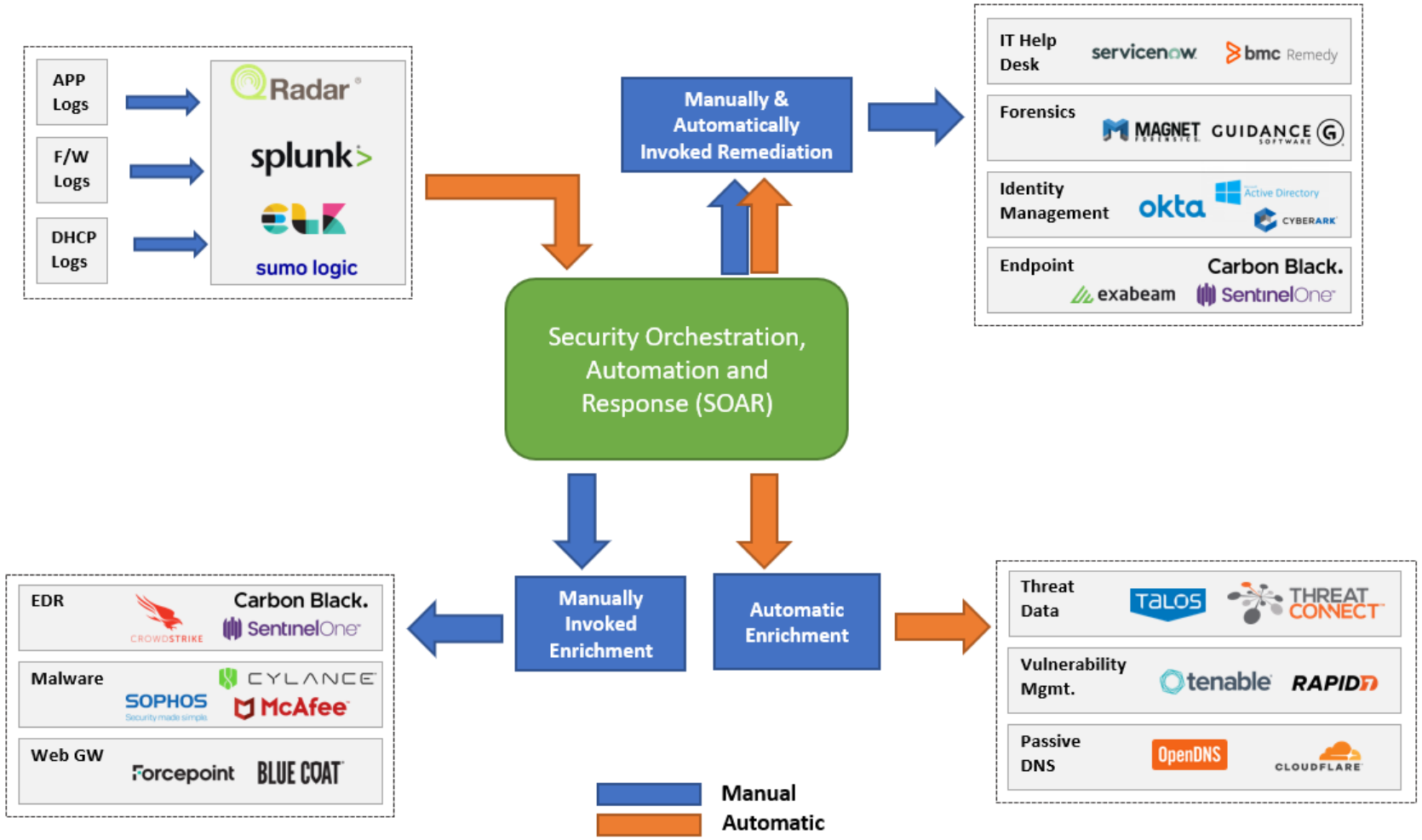
- Automating malware analysis
- Provision or deprovision new users
- Query logs for further critical data

Response

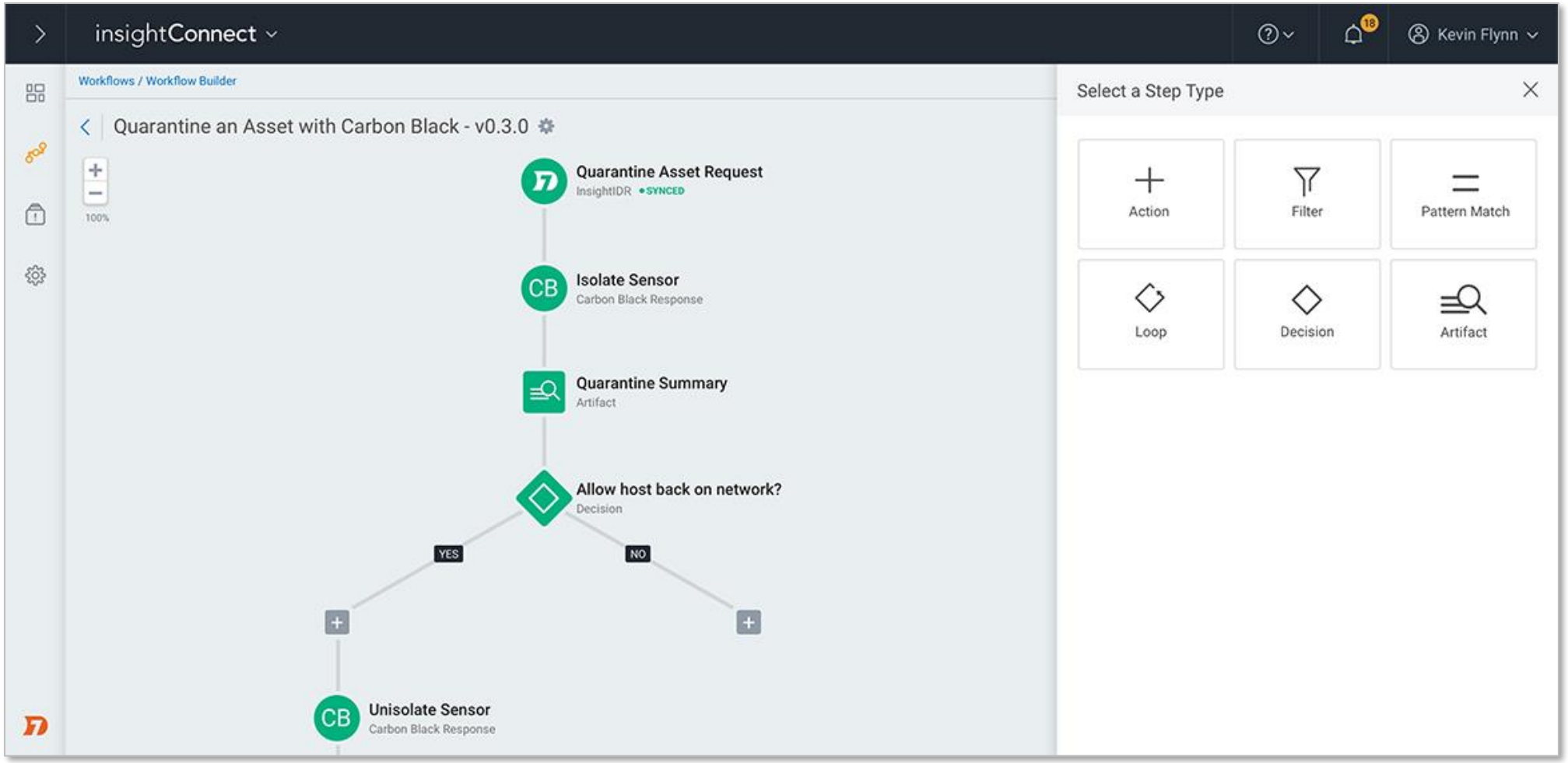
- Helps analysts to manage security incidents, collaborate and share data for incident resolution
- Assist with alert triage and processing, case and threat management.

- Collect data from other security tools (i.e. SIEM)
- Submit a ticket to case management system
- Convert incident data into threat intelligence

Incident Response| SOAR Architecture



Incident Response| Typical SOAR Environment (e.g. Rapid7)



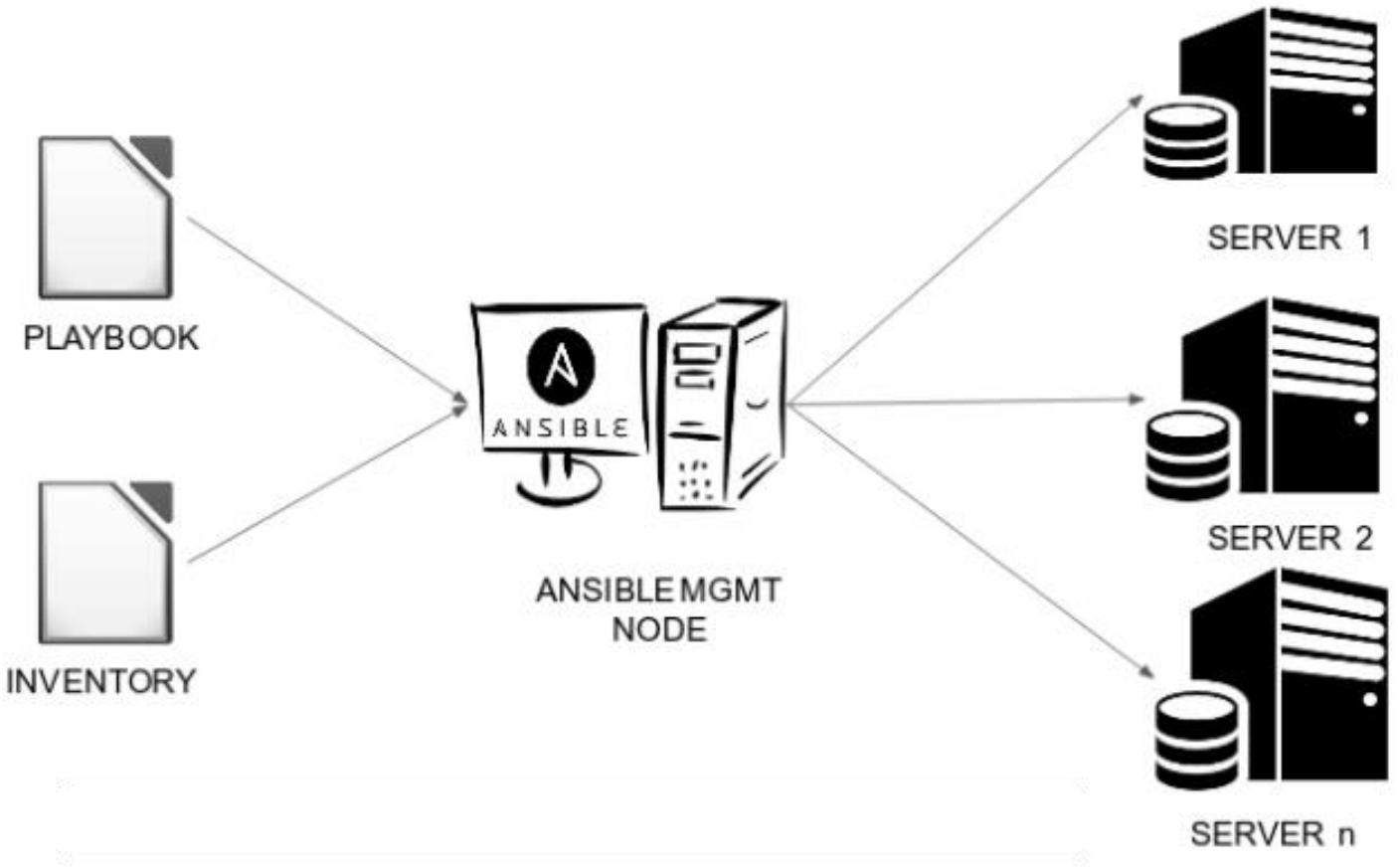
Incident Response | Automate the Process

- Environments where enterprise tools are not deployed (i.e. EDR)
- Legacy environments
- ICS / OT environments (i.e. SCADA/DCS)
- Large cloud environments (i.e. hundreds of hosts)
- Recently acquired network environments (M&A)
- Lack of proper centralized log collection
- Air-gapped networks
- **We know, environments aren't always what we want them to be!**

Why Ansible? | Incident Response

- Not really a security tool / used for configuration management
- Commonly used in DevOps environments
- Agentless
- Python-based
- SSH/Windows Remote Management
- Extensible and modular
- Push-based architecture
- Also supports management of network and storage devices
- **Easy adoption**

Ansible Architecture | Incident Response



Source: Medium.com

Ansible Playbook | Incident Response

```
---
 - name: Playbook
   hosts: webservers
   become: yes
   become_user: root
   tasks:
    - name: ensure apache is latest version
      yum:
        name: httpd
        state: latest
    - name: ensure apache is running
      service:
        name: httpd
        state: started
```

Playbook Name

Group of Hosts

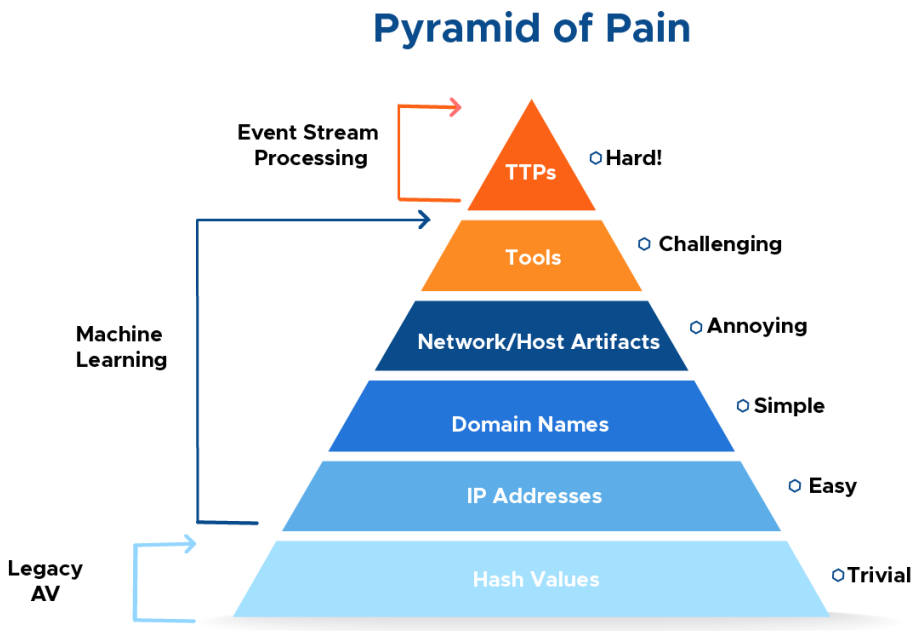
Executed with elevated privileged (su)

Set of tasks to execute, All tasks would be defined below this

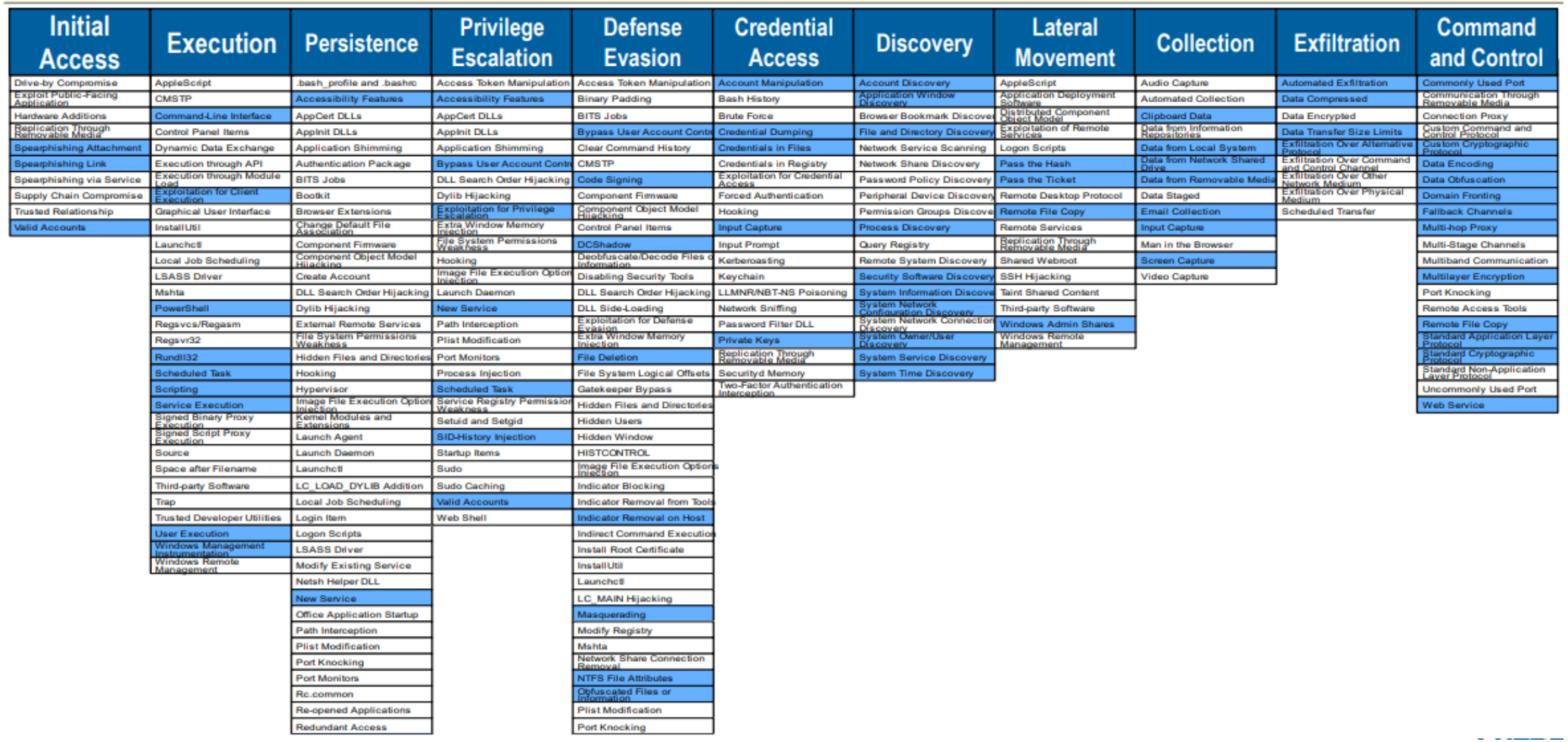
- Playbook – YAML file
  - Play – defines a set of activities (tasks) to be run on hosts
  - Task – an action to be performed on the host
    - Execute a command
    - Run a script
    - Install a package
    - Shutdown/Restart host or service

Automating Playbooks| ATT&CK Framework

- Using the ATT&CK framework to help build your playbooks and understand where to apply automation
- Remember the pyramid of pain – you are looking for TTPs or ATT&CK techniques
- Translate these techniques into steps in your playbooks
- If a technique identifies that the attacker will schedule a task/job, maybe from a containment perspective, you want to identify where a task was created and remove it.



Automating Playbooks| ATT&CK Framework



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	System Time Discovery				Standard Non-Application Layer Protocol
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	Hidden Files and Directories						Web Service
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window						
	Source	Launch Daemon	Startup Items	HISTCONTROL						
	Space after Filename	Launchctl	Sudo	Image File Execution Options Injection						
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking						
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools						
	Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host						
	User Execution	Logon Scripts		Indirect Command Execution						
	Windows Management Instrumentation	LSASS Driver		Install Root Certificate						
	Windows Remote Management	Modify Existing Service		InstallUtil						
		Netsh Helper DLL		Launchctl						
		New Service		LC_MAIN Hijacking						
		Office Application Startup		Masquerading						
		Path Interception		Modify Registry						
		Plist Modification		Mshta						
		Port Knocking		Network Share Connection Removal						
		Port Monitors		NTFS File Attributes						
		Rc.common		Obfuscated Files or Information						
		Re-opened Applications		Plist Modification						
		Redundant Access		Port Knocking						

Ansible Use Cases | Triage + Containment Examples

Data Collection (Triage)

- Running Processes
- Netstat
- Memory Dump
- Apache Logs
- System Logs
- Bash History
- Web Server Files (webdir)
- Network device configurations

System Interaction (Containment)

- Reset passwords
- Create / disable user accounts
- Start/Stop Services
- Edit host-based firewall rules (i.e. firewalld/iptables)
- Enable and disable Windows features
- Manage and install Windows updates
- Large scale config changes on network devices
- Shut down interfaces on network devices

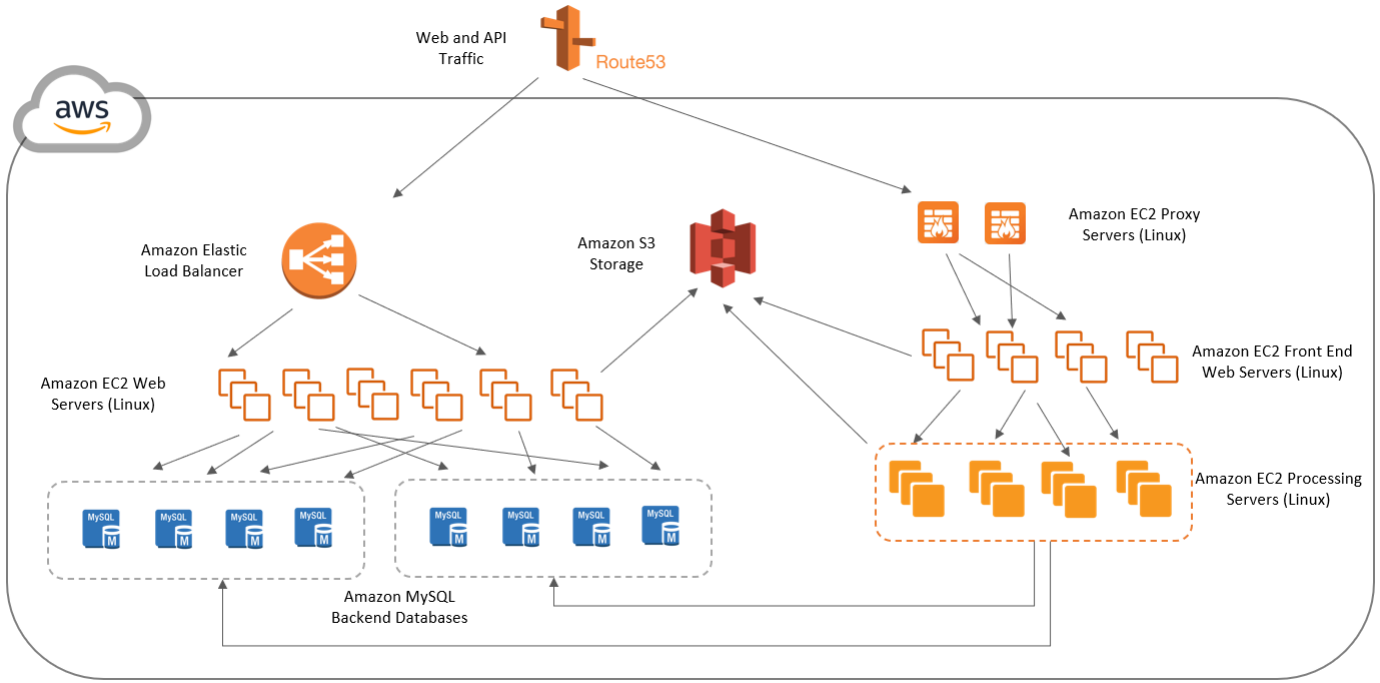
Automating Playbooks| ATT&CK Technique

Technique	New Service
Description	When operating systems boot up, they can start programs or applications called services that perform background system functions. […] Adversaries may install a new service which will be executed at startup by directly modifying the registry or by using tools.
Platform	Windows
Permissions Required	Administrator, SYSTEM
Effective Permissions	SYSTEM
Detection	Monitor service creation through changes in the Registry and common utilities using command-line invocation …
Mitigation	Limit privileges of user accounts and remediate Privilege Escalation vectors…
Data Sources	Windows registry, process monitoring, command-line parameters
Examples	Carbanak, Lazarus Group, TinyZBot, Duqu, CozyCar, CosmicDuke, hcdLoader, …
References	1. Microsoft. (n.d.). Services. Retrieved June 7, 2016.

Scenario| Insider Threat Access

MediTech Pharmaceuticals Corporation

BioLife *Vaccines* PharmaceutIcals Inc.



BioLife Vaccines COVID-19 Research Environment

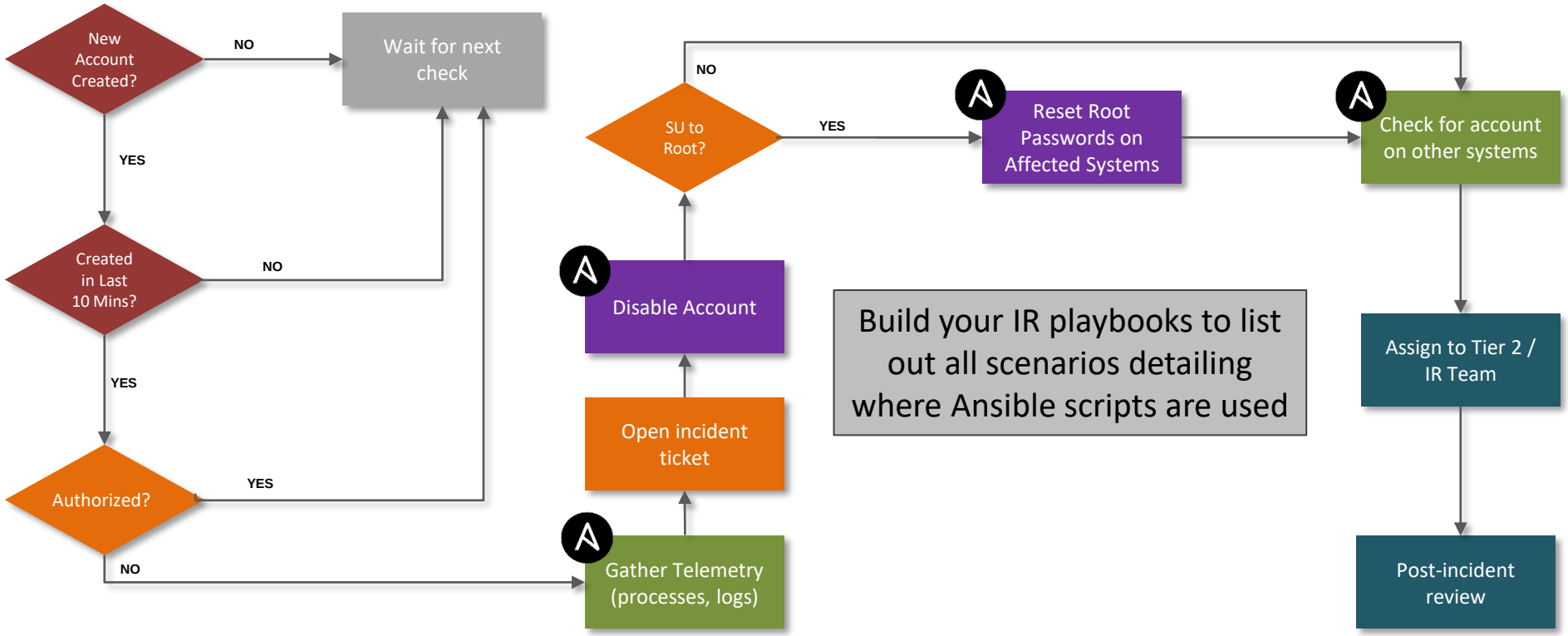
- Minimal knowledge as many were laid off
- Quickly built during the pandemic
- Flat network
- No EDR
- No centralized authentication / access via SSH (with keys)
- 100s of Linux-based operating system / Apache Webservers

Scenario| Insider Threat Access

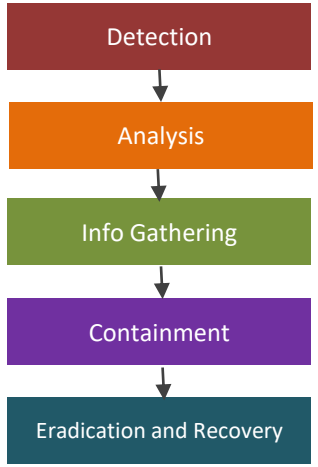
Stephen Smith

- Systems Administration
- Been with MediTech for almost 10 years
- Disgruntled, was past up for a director position
- Opportunity with the BioLife acquisition
- His plan is to create an number of backdoor accounts on some key Linux servers hosted in the cloud that store IP for MediTech.
- Then install a vulnerability creating a backdoor in the webserver to access the data from outside the network.

Typical Playbook | Insider Threat Access



Build your IR playbooks to list out all scenarios detailing where Ansible scripts are used



Ansible Use Cases | Does Account Exist?

- We have identified a backdoor account our insider has been using
- Let's check to see if he has created it on any of our servers

```
---
- name: Check for User Account
  hosts: all
  tasks:
  - name: Check for User Account
    become: yes
    become_user: root
    register: presence
    shell: "grep -i 'jsmith' /etc/passwd"

  - name: "Does User Account Exist"
    debug: msg="User account exists"
    when: presence is changed
```

```
root@ansible-control:/home/admin/playbooks
[root@ansible-control playbooks]# ansible-playbook UserExists.yml

PLAY [Check for User Account] **************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.2.141]
ok: [192.168.2.140]

TASK [Check for User Account] **************************************************
fatal: [192.168.2.141]: FAILED! => {"changed": true, "cmd": "grep -i 'jsmith'
"msg": "non-zero return code", "rc": 1, "start": "2021-03-02 11:55:18.912184"
changed: [192.168.2.140]

TASK [Does User Account Exist] *************************************************
ok: [192.168.2.140] => {
    "msg": "User account exists"
}

PLAY RECAP *********************************************************************
192.168.2.140              : ok=3    changed=1    unreachable=0    failed=0
192.168.2.141              : ok=1    changed=0    unreachable=0    failed=1

[root@ansible-control playbooks]#
```

Ansible Use Cases | Disable Account

- Now that we have identified the account, we want to disable it
- We don't want to remove it, as there may be forensics information available
- **Usermod** command will lock and disable user account
- We can then ingest the secure log to see whenever our insider attempts to login with the backdoor account

```
---
- name: Disable User Account
  hosts: all
  tasks:
  - name: Disable User Account
    become: yes
    become_user: root
    shell: "usermod -L -e 1 jsmith"
```

root@ansible-control:/home/admin/playbooks

```
[root@ansible-control playbooks]# ansible-playbook DisableAccount.yml

PLAY [Disable User Account] ****************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.2.141]
ok: [192.168.2.140]

TASK [Disable User Account] ****************************************************
fatal: [192.168.2.141]: FAILED! => {"changed": true, "cmd": "usermod -L -e 1 jsmit
 "non-zero return code", "rc": 6, "start": "2021-03-02 12:14:54.884604", "stderr":
'jsmith' does not exist"], "stdout": "", "stdout_lines": []}
changed: [192.168.2.140]

PLAY RECAP *********************************************************************
192.168.2.140              : ok=2    changed=1    unreachable=0    failed=0    ski
192.168.2.141              : ok=1    changed=0    unreachable=0    failed=1    ski
```

root@ansible-node01:~

```
[root@ansible-node01 ~]# ssh -l jsmith 192.168.2.140
jsmith@192.168.2.140's password:
Your account has expired; please contact your system administrator
Authentication failed.
[root@ansible-node01 ~]#
```

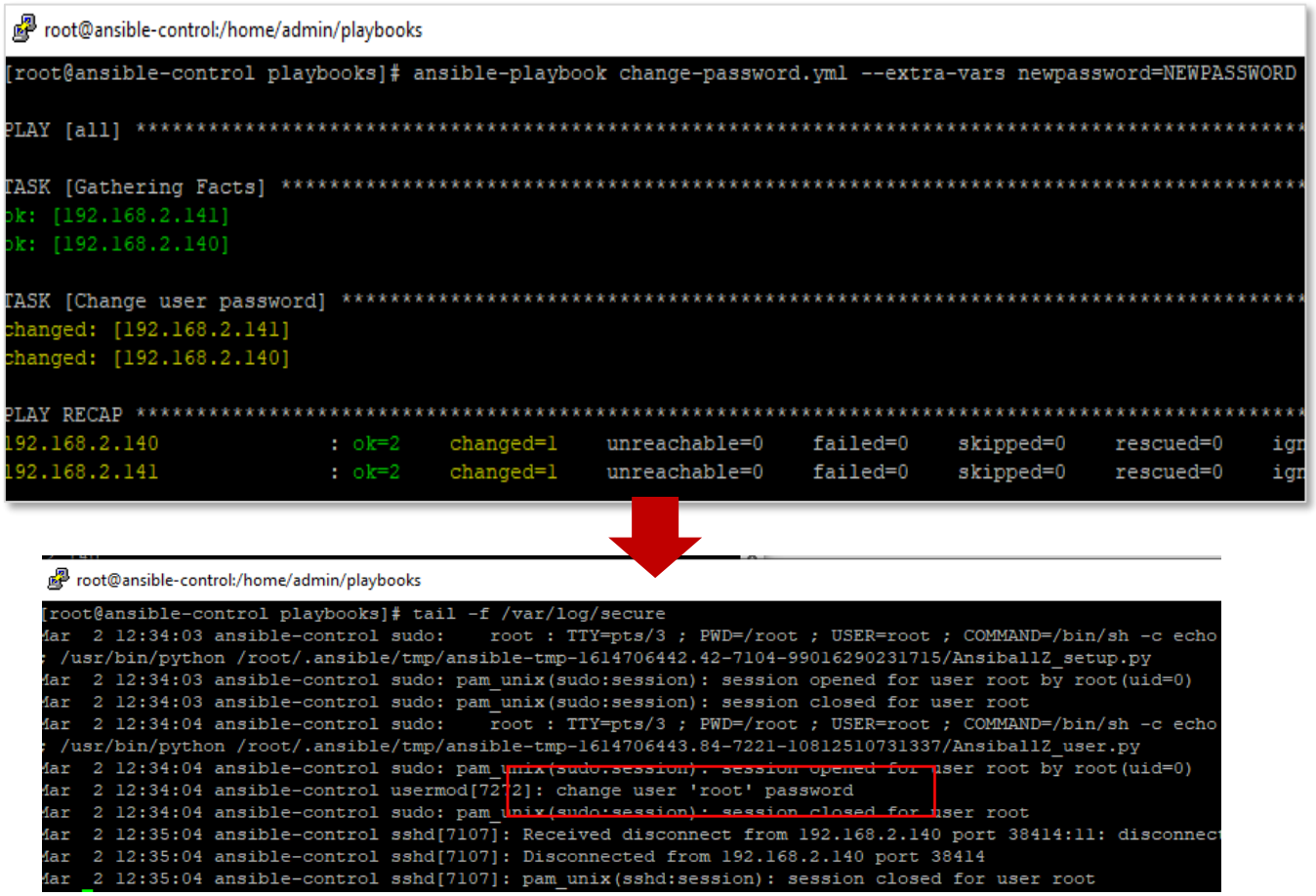
root@ansible-control:/home/admin/playbooks

```
[root@ansible-control playbooks]# tail -5 /var/log/secure
Mar  2 12:16:45 ansible-control passwd: pam_unix(passwd:chauthtok): password changed for jsmith
Mar  2 12:16:59 ansible-control sshd[6873]: Connection closed by 192.168.2.141 port 38060 [preauth]
Mar  2 12:17:06 ansible-control sshd[6875]: pam_unix(sshd:account): account jsmith has expired (account expired)
Mar  2 12:17:06 ansible-control sshd[6875]: Failed password for jsmith from 192.168.2.141 port 38062 ssh2
Mar  2 12:17:06 ansible-control sshd[6875]: fatal: Access denied for user jsmith by PAM account configuration [preauth]
```

Ansible Use Cases | Mass Password Change

- If we think the root password has been compromised
- We want to do a mass password change
- We can set the password at the command line without having to embed it in a playbook

```
---
- hosts: all
  become: yes
  become_user: root
  tasks:
    - name: Change user password
      user:
        name: root
        update_password: always
        password: "{{ newpassword|password_hash('sha512') }}"
```



ansible-playbook change-password.yml --extra-vars newpassword=**NEWPASSWORD**

Ansible Automation| Other Helpful Use Cases

Ansible Use Cases | Pulling Logs/Data from Remote Hosts

Create a local evidence directory on my Ansible Control Server

```
---
- name: Create Triage Directory Locally
  hosts: all
  connection: local

  tasks:
  - name: Make evidence collection directory ($pwd/artifacts)
    file:
      path: artifacts/{{ inventory_hostname }}
      state: directory
      recurse: yes
```

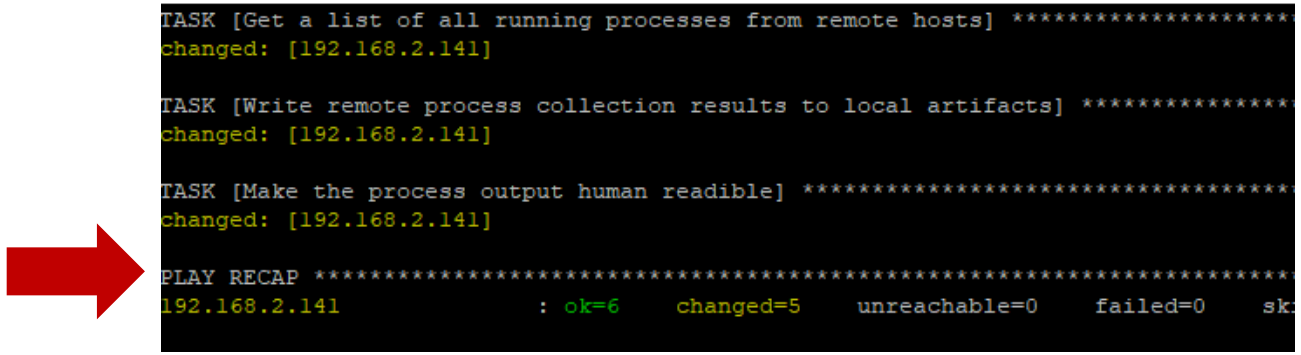
- We want to run this action prior to any further playbook to ensure the destination directory exists – based on the updated inventory
- You can even run this in a cron job.

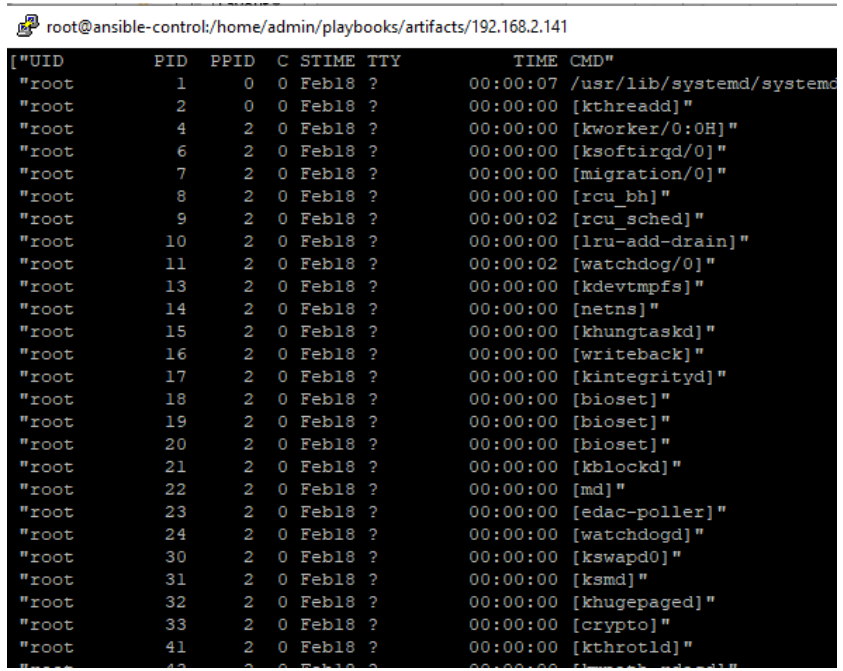
Ansible Use Cases | Process Listing

```
- name: Get a list of all running processes from remote hosts
  shell: ps -ef
  register: ps_result

- name: Write remote process collection results to local artifacts
  local_action:
    module: copy
    content: "{{ ps_result.stdout_lines }}"
    dest: artifacts/{{ inventory_hostname }}/processlist-
{{ansible_date_time.iso8601}}.txt

- name: Make the process output human readible
  local_action:
    module: replace
    path: artifacts/{{ inventory_hostname }}/processlist-
{{ansible_date_time.iso8601}}.txt
    before: ','
    regexp: ','
    replace: '\n'
```





Ansible Use Cases | Pull Apache Logs

```
- name: List apache log files in /var/log/apache2
  shell: (cd /var/log/httpd; find . -maxdepth 1 -type f) | cut -d'/' -f2
  register: wwwlogs_to_copy

- name: Download apache log files to artifacts on localhost
  fetch:
    src: /var/log/httpd/{{ item }}
    dest: artifacts/
  with_items:
    - "{{ wwwlogs_to_copy.stdout_lines }}"
```

root@ansible-control:/home/admin/playbooks/artifacts/192.168.2.141

```
TASK [List apache log files in /var/log/httpd] *********************************
changed: [192.168.2.141]

TASK [Download apache log files to artifacts on localhost] *********************
changed: [192.168.2.141] => (item=error_log)
changed: [192.168.2.141] => (item=access_log)

PLAY RECAP *********************************************************************
192.168.2.141              : ok=6    changed=5    unreachable=0    failed=0
```

root@ansible-control:/home/admin/playbooks/artifacts/192.168.2.141/var/log/httpd

```
[root@ansible-control httpd]# ls -l
total 44
-rw-r--r--. 1 root root 34208 Feb 19 10:44 access_log
-rw-r--r--. 1 root root  7896 Feb 19 10:44 error_log
[root@ansible-control httpd]# more access_log
192.168.2.254 - - [19/Feb/2021:09:20:15 -0500] "GET / HTTP/1.1" 403 4897 "-" "Mozilla/5.0 (Windows NT
192.168.2.254 - - [19/Feb/2021:09:20:15 -0500] "GET /noindex/css/bootstrap.min.css HTTP/1.1" 200 19341
.0.4324.182 Safari/537.36"
192.168.2.254 - - [19/Feb/2021:09:20:15 -0500] "GET /noindex/css/open-sans.css HTTP/1.1" 200 5081 "htt
24.182 Safari/537.36"
192.168.2.254 - - [19/Feb/2021:09:20:15 -0500] "GET /images/apache_pb.gif HTTP/1.1" 200 2326 "http://1
2 Safari/537.36"
192.168.2.254 - - [19/Feb/2021:09:20:15 -0500] "GET /images/poweredby.png HTTP/1.1" 200 3956 "http://1
2 Safari/537.36"
192.168.2.254 - - [19/Feb/2021:09:20:15 -0500] "GET /noindex/css/fonts/Bold/OpenSans-Bold.woff HTTP/1.
37.36 (KHTML, like Gecko) Chrome/88.0.4324.182 Safari/537.36"
192.168.2.254 - - [19/Feb/2021:09:20:15 -0500] "GET /noindex/css/fonts/Light/OpenSans-Light.woff HTTP/
/537.36 (KHTML, like Gecko) Chrome/88.0.4324.182 Safari/537.36"
192.168.2.254 - - [19/Feb/2021:09:20:15 -0500] "GET /noindex/css/fonts/Bold/OpenSans-Bold.ttf HTTP/1.1
7.36 (KHTML, like Gecko) Chrome/88.0.4324.182 Safari/537.36"
```

Ansible Use Cases | Service Stop

```
[SCADAServers]
SCADA[99:101]-node.example.com

[dbServers]
 db01.intranet.mydomain.net
 db02.intranet.mydomain.net
 db03.intranet.mydomain.net

[webservers]
192.168.2.141
```

ansible *webservers* -m service -a "name=httpd state=stopped"

```
root@ansible-control:/home/admin/playbooks
[root@ansible-control playbooks]# ansible webservers -m service -a "name=httpd state=stopped"
192.168.2.140 | FAILED! => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "msg": "Could not find the requested service httpd: host"
}
192.168.2.141 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "name": "httpd",
    "state": "stopped",
    "status": {
        "ActiveEnterTimestamp": "Tue 2021-03-02 12:43:52 EST",
        "ActiveEnterTimestampMonotonic": "122367290414",
        "ActiveExitTimestamp": "Tue 2021-03-02 12:43:51 EST",
        "ActiveExitTimestampMonotonic": "122366196871",
        "ActiveState": "active",
        "After": "tmp.mount -.mount basic.target systemd-journald.socket system.slice nss-look
        "AllowIsolate": "no",
        "AmbientCapabilities": "0",
        "AssertResult": "yes",
        "AssertTimestamp": "Tue 2021-03-02 12:43:52 EST",
```

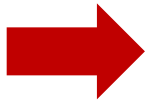
Ansible Use Cases | Firewall Rules

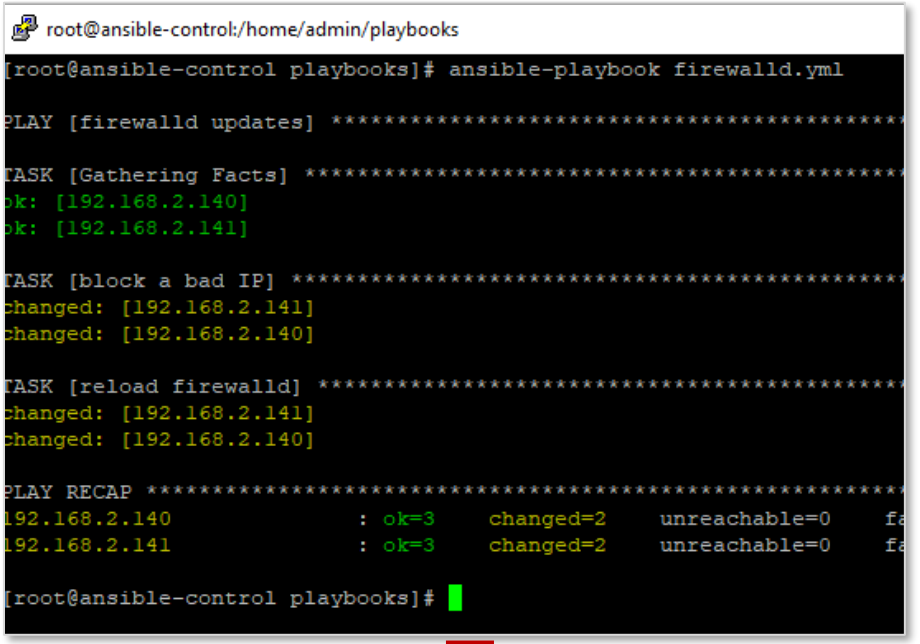
- In the event that you need to block known “bad” IP addresses
- Allows you to push out firewalld rules to a large number of hosts

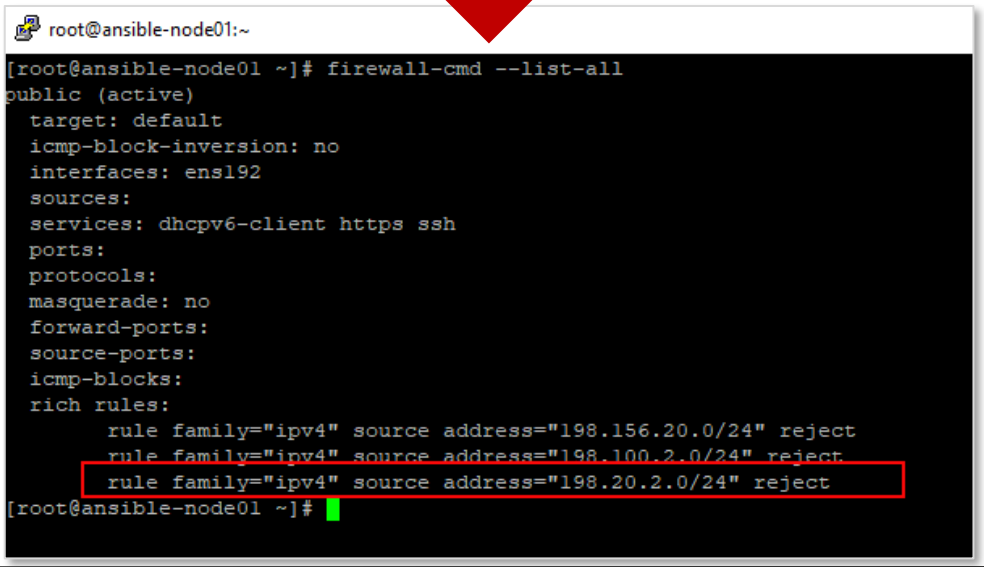
```
---
- name: firewalld Rule Update
  hosts: all
  become: yes

  tasks:
  - name: Block a Bad Block of IP Addresses
    firewalld:
      zone: public
      rich_rule: rule family=ipv4 source address=198.20.2.0/24 reject
      permanent: yes
      state: enabled

  - name: reload firewalld service
    service:
      name: firewalld
      state: restarted
```







In Closing…

Don’t forget how critical the triage and containment phases of the incident response process are – they are incredibly critical in **reducing the dwell time**.

Remember the **important role** that **automation** can plan in your IR plan – it adds **speed** to your containment when you are dealing with a large number of hosts.

I have only shown you what **Ansible** can do with **Linux**. As you may recall it can support a number of other platforms including **firewalls and network devices**.

Mapping your triage and containment playbooks to the **MITRE ATT&CK** framework will help you ensure that your processes reflect **actual adversary TTPs**.

Peter Morin

petermorin123@gmail.com

Twitter: @PeterMorin123

http://www.petermorin.com