EDR Remote Execution - Using Respond Console Manual

Research on Elastic EDR Response Actions for:

1. Forensic commands for malware investigation on isolated hosts
2. API integration documentation for external systems

Key Findings:

- Elastic EDR has 11 response actions for remote host management
- Primary tool: `execute` command for running forensic commands
- Full API support available for integration with external systems
- Can automate investigation and remediation workflows

AVAILABLE RESPONSE ACTIONS

Action	Purpose	Use Case
isolate	Block host from network	Contain infected host
release	Restore network access	Release clean host
status	Get host information	Check agent status
processes	List running processes	Find malicious processes
kill-process	Terminate process	Stop malware
suspend-process	Pause process	Freeze for analysis
get-file	Download file from host	Extract malware samples
upload	Upload file to host	Deploy remediation tools
execute	Run shell commands	Main forensic investigation tool
scan	Scan for malware	Verify system clean
runscript	Run scripts	Third-party EDR only

Source: https://www.elastic.co/guide/en/security/8.18/response-actions.html

FORENSIC COMMANDS FOR INFECTED HOSTS

Investigation

Windows Commands:

```
Windows Commands:

# Network connections (find C2 servers)
execute --command "netstat -ano" --timeout 30s

# Running processes
execute --command "tasklist /v /fo csv" --timeout 30s

# Scheduled tasks (persistence check)
execute --command "schtasks /query /fo csv /v" --timeout 60s

# Startup programs
execute --command "wmic startup get Caption,Command" --timeout 30s

# DNS cache (domains contacted)
execute --command "ipconfig /displaydns" --timeout 30s

# PowerShell history
execute --command "type
%APPDATA%\\Microsoft\\Windows\\PowerShell\\PSReadLine\\ConsoleHost_history.txt" --timeout 10s

# Registry persistence keys
execute --command "reg query HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run" --
timeout 10s
```

Linux Commands:

```
Linux Commands:

# Network connections
execute --command "netstat -tulpn" --timeout 30s
```

```
# Process list
execute --command "ps auxf" --timeout 30s

# Cron jobs (persistence)
execute --command "crontab -l && cat /etc/crontab" --timeout 30s

# Bash history
execute --command "cat ~/.bash_history" --timeout 10s

# SSH keys
execute --command "cat ~/.ssh/authorized_keys" --timeout 10s

# Running services
execute --command "systemctl list-units --type=service --state=running" --timeout 30s
```

What to Look For:

- Network connections to unknown foreign IPs
- Processes running from temp directories
- Scheduled tasks with suspicious scripts
- Unknown startup programs
- Suspicious PowerShell/bash commands in history
- Registry entries pointing to malware

Extract Evidence

```
# Get suspicious file
get-file --path "C:\\Users\\Public\\suspicious.exe" --comment "Extract malware sample"

# Get logs
get-file --path "C:\\Windows\\System32\\winevt\\Logs\\Security.evtx" --comment "Get security logs"
```

Note: Files are downloaded as password-protected .zip (password: elastic)
Max file size: 100 MB

Remediation

```
# Kill malicious process
kill-process --pid 1234 --comment "Terminate malware"

# Delete malware file (Windows)
```

```
execute --command "del /F /Q C:\\path\\to\\malware.exe" --timeout 10s

# Delete malware file (Linux)
execute --command "rm -f /path/to/malware" --timeout 10s

# Remove registry persistence (Windows)
execute --command "reg delete HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run /v
MalwareName /f" --timeout 10s

# Remove scheduled task (Windows)
execute --command "schtasks /delete /tn MaliciousTask /f" --timeout 10s

# Scan system
scan --path "C:\\" --comment "Full system scan"
```

Verification and Release

```
# Verify processes clean
processes

# Verify no malicious connections
execute --command "netstat -ano" --timeout 30s

# If clean, release from isolation
release --comment "Host verified clean"
```

Investigation Workflow

```
1. ISOLATE → isolate --comment "Malware detected"
2. INVESTIGATE → Run forensic commands above
3. EXTRACT → get-file for suspicious files/logs
4. REMEDIATE → kill-process, delete files, remove persistence
5. SCAN → scan --path to verify clean
6. VERIFY → Re-check processes and connections
7. RELEASE → release --comment "System clean"
```

API INTEGRATION

API Overview
API Exists: YES - Full REST API support
Base URL: example: `https://<kibana-host>:5601`
Authentication: API Key (recommended) or Basic Auth

Content-Type: application/json

Required Headers:

```
Authorization: ApiKey <base64-encoded-key>
Content-Type: application/json
kbn-xsrf: true
```

Source: https://www.elastic.co/guide/en/security/current/response-actions-api.html

API Endpoints

All response actions can be triggered via API:

```
POST /api/endpoint/action/{action_type}
```

Available action types:

- isolate
- unisolate (release)
- running-processes (processes)
- kill-process
- suspend-process
- get-file
- execute
- upload
- scan

Example: Execute Command via API

Request:

```
POST https://kibana.example.com:5601/api/endpoint/action/execute
Authorization: ApiKey <your-api-key>
Content-Type: application/json
kbn-xsrf: true

{
  "endpoint_ids": ["host-agent-id-here"],
  "parameters": {
    "command": "netstat -ano",
    "timeout": 30
  },
  "comment": "Check network connections"
```

```
}
```

Response:

```
{
  "data": {
    "id": "action-12345-abcd",
    "status": "pending",
    "command": "execute",
    "agents": ["host-agent-id-here"],
    "startedAt": "2024-12-10T10:30:00.000Z",
    "isCompleted": false
  }
}
```

Example: Isolate Host via API

Request:

```
POST https://kibana.example.com:5601/api/endpoint/action/isolate
Authorization: ApiKey <your-api-key>
Content-Type: application/json
kbn-xsrf: true

{
  "endpoint_ids": ["host-agent-id-here"],
  "comment": "Malware detected - isolating host"
}
```

Response:

```
{
  "data": {
    "id": "action-67890-efgh",
    "status": "pending",
    "command": "isolate",
    "agents": ["host-agent-id-here"]
  }
}
```

Check Action Status

Request:

```
GET https://kibana.example.com:5601/api/endpoint/action/{action_id}
Authorization: ApiKey <your-api-key>
kbn-xsrf: true
```

Response:

```
{
  "data": {
    "id": "action-12345-abcd",
    "status": "successful",
    "command": "execute",
    "isCompleted": true,
    "outputs": {
      "host-agent-id": {
        "type": "json",
        "content": {
          "output": "command output here..."
        }
      }
    }
  }
}
```

API Integration Benefits

1. **Automation** - Trigger actions programmatically without manual intervention
2. **Integration** - Connect Elastic EDR with external SIEM, ticketing systems, or custom tools
3. **Bulk Operations** - Execute commands on multiple hosts simultaneously
4. **Custom Workflows** - Build automated investigation and response playbooks
5. **Faster Response** - Reduce response time from minutes to seconds

Python Example Code

```
import requests

# Configuration
KIBANA_URL = "https://kibana.example.com:5601"
API_KEY = "your-base64-encoded-api-key"

headers = {
```

```
    "Authorization": f"ApiKey {API_KEY}",
    "Content-Type": "application/json",
    "kbn-xsrf": "true"
}

# Isolate host
def isolate_host(endpoint_id, comment):
    url = f"{KIBANA_URL}/api/endpoint/action/isolate"
    payload = {
        "endpoint_ids": [endpoint_id],
        "comment": comment
    }
    response = requests.post(url, headers=headers, json=payload)
    return response.json()

# Execute command
def execute_command(endpoint_id, command, timeout=600):
    url = f"{KIBANA_URL}/api/endpoint/action/execute"
    payload = {
        "endpoint_ids": [endpoint_id],
        "parameters": {
            "command": command,
            "timeout": timeout
        },
        "comment": "Automated forensic command"
    }
    response = requests.post(url, headers=headers, json=payload)
    return response.json()

# Usage
result = isolate_host("abc-123-endpoint-id", "Malware detected")
print(f"Action ID: {result['data']['id']}")

result = execute_command("abc-123-endpoint-id", "netstat -ano")
print(f"Action ID: {result['data']['id']}")
```

Prerequisites for API Integration

1. **API Key** - Create in Kibana: Stack Management → API Keys
2. **Required Privileges:**
   - Host Isolation

- Process Operations
- Execute Operations
- File Operations
- Scan Operations

3. **Network Access** - System must reach Kibana URL (port 5601)
4. **Endpoint IDs** - Map hostnames to Elastic endpoint agent IDs

API Action Mapping Table

Response Action	API Endpoint	Required Parameters
Isolate	/api/endpoint/action/isolate	endpoint_ids
Release	/api/endpoint/action/unisolate	endpoint_ids
Get Processes	/api/endpoint/action/running-processes	endpoint_ids
Execute Command	/api/endpoint/action/execute	endpoint_ids, parameters.command
Kill Process	/api/endpoint/action/kill-process	endpoint_ids, parameters.pid
Get File	/api/endpoint/action/get-file	endpoint_ids, parameters.path
Scan	/api/endpoint/action/scan	endpoint_ids, parameters.path

ADDITIONAL REFERENCES

Official Documentation:

- Response Actions Overview: https://www.elastic.co/guide/en/security/8.18/response-actions.html
- Response Actions API: https://www.elastic.co/guide/en/security/current/response-actions-api.html
- Execute API: https://www.elastic.co/guide/en/security/current/execute-api.html
- API Key Management: https://www.elastic.co/guide/en/kibana/current/api-keys.html

Security APIs:

- Elastic Security APIs: https://www.elastic.co/guide/en/security/current/security-apis.html

Revision #1
Created 12 December 2025 06:13:47
Updated 12 December 2025 06:31:26