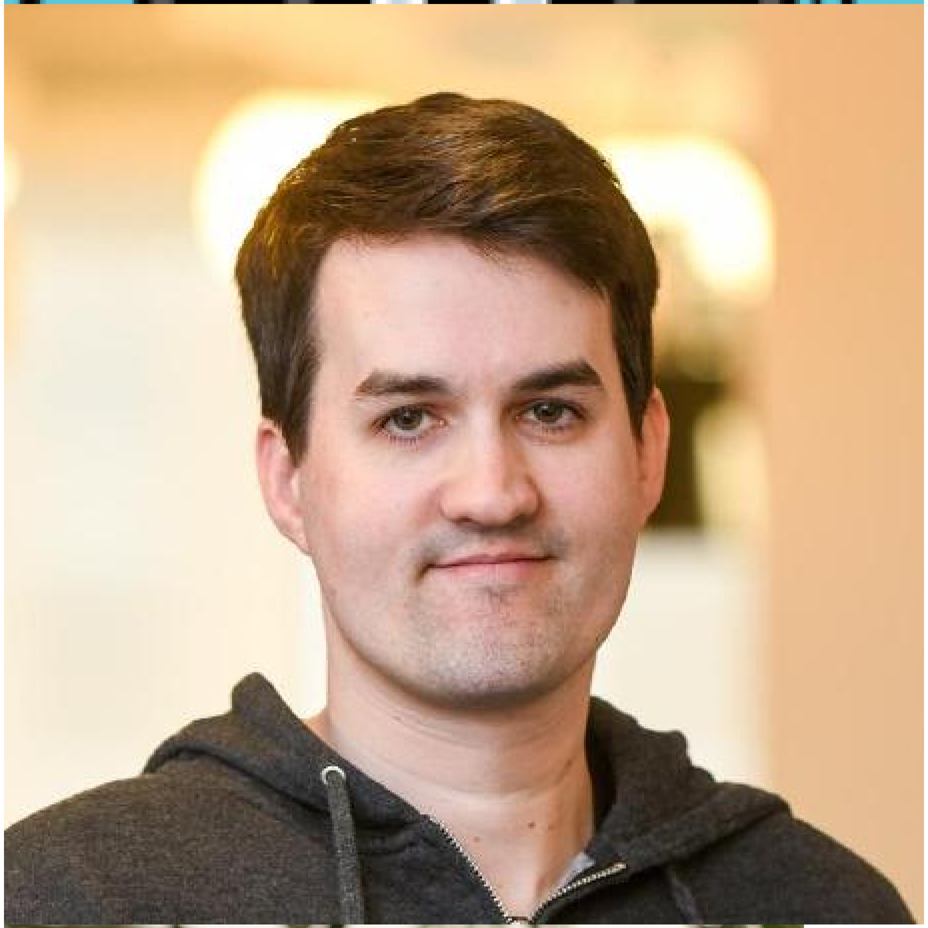
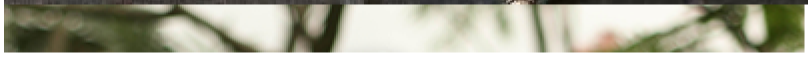
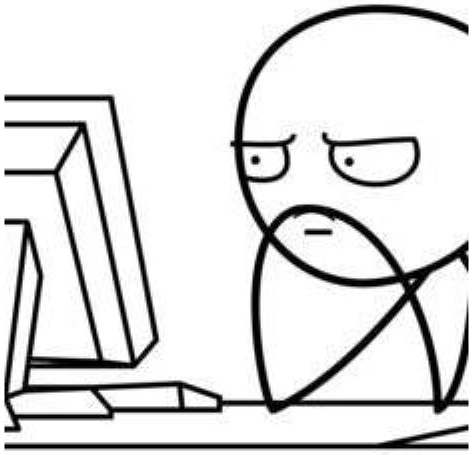
Collecting Cobalt Strike Beacons with the Elastic Stack

elastic.github.io/security-research/intelligence/2022/01/02.collecting-cobalt-strike-beacons/article/

Elastic Security Research
Collecting Cobalt Strike Beacons with the Elastic
Part 1 - Processes and technology needed to extract Cobalt Strike implant beacons

Tutorial Cobalt Strike Fleet



2022-01-19

Overview¶

Cobalt Strike is a premium offensive security tool leveraged by penetration testers and red team members as a way to emulate adversary behavior. The goal is to validate security detection capabilities and processes replicating a real-world intrusion. While Cobalt Strike is a legitimate tool, it is often abused by actual threat actors as a way to gain and maintain persistence into targeted networks.

To manage command and control, Cobalt Strike leverages an implant that uses beacon configuration known as a Malleable Command and Control (Malleable C2) profile. A Malleable C2 profile contains a tremendous number of options to configure the beacon's functionality, please see Cobalt Strike's official documentation for specifics on configuring Malleable C2 beacons.

This blog will focus on using the Elastic Stack to collect Cobalt Strike beacon payloads, extract and parse the beacon configurations, and an analysis of the metadata within the configurations. This will all be taken from the memory of targeted Windows endpoints that we've collected from our telemetry.

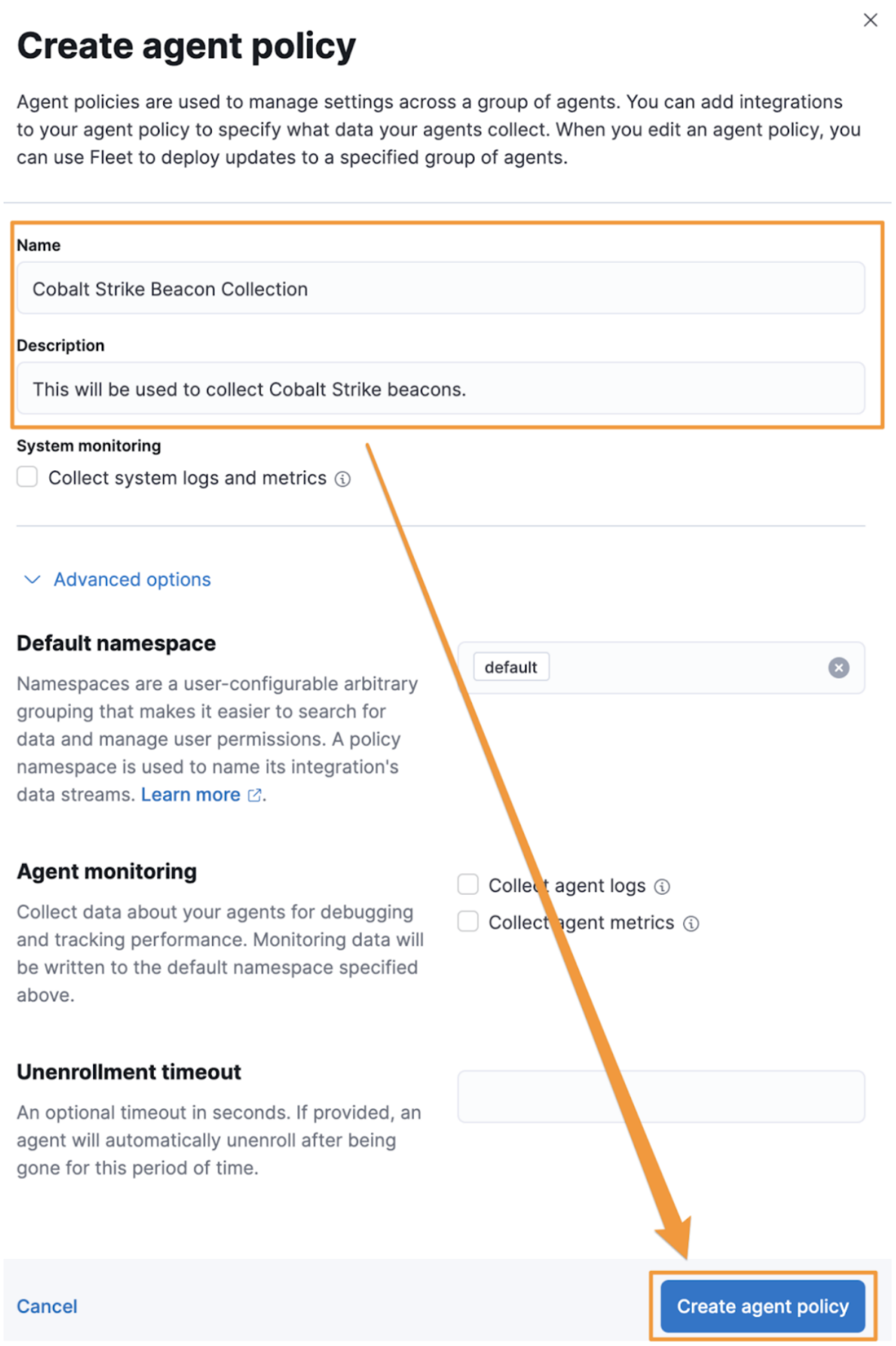
The Fleet Policy¶

Fleet is an app in Kibana that provides a central place to configure and monitor your Elastic Agents. Fleet uses integrations, which are unified plugins that allow data to be collected from apps and services, and then stored in Elasticsearch. Integrations are added to policies, and

Elastic Agents are added to policies.

First, we need to configure the collection of shellcode and malicious memory regions in a Fleet policy. This will collect 4MB of data from memory surrounding shellcode and malicious memory events. It should be noted that this collection may significantly increase the amount of data stored in Elasticsearch.

You can add this to an existing policy or create a new policy. To create a new policy, in Kibana, navigate to Fleet → Agent Policies → Create agent policy. Give your policy a name and description. Optionally, you can disable “System monitoring” and “Agent monitoring” to reduce the amount of system and agent metadata collected from your endpoints. Click on “Create agent policy”.


Create agent policy

Agent policies are used to manage settings across a group of agents. You can add integrations to your agent policy to specify what data your agents collect. When you edit an agent policy, you can use Fleet to deploy updates to a specified group of agents.

Name

Cobalt Strike Beacon Collection

Description

This will be used to collect Cobalt Strike beacons.

System monitoring

☐ Collect system logs and metrics

Advanced options

Default namespace

Namespaces are a user-configurable arbitrary grouping that makes it easier to search for data and manage user permissions. A policy namespace is used to name its integration's data streams. Learn more.

default

Agent monitoring

Collect data about your agents for debugging and tracking performance. Monitoring data will be written to the default namespace specified above.

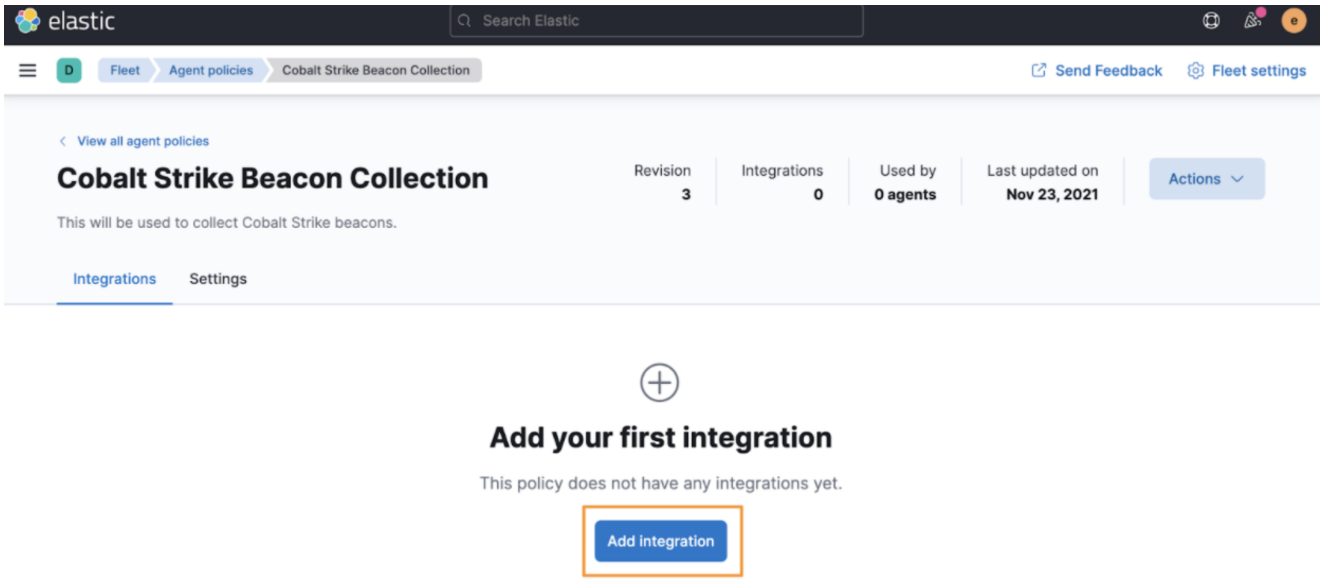
☐ Collect agent logs

☐ Collect agent metrics

Unenrollment timeout

An optional timeout in seconds. If provided, an agent will automatically unenroll after being gone for this period of time.

Next, click on your new policy and click the “Add integration button.



Cobalt Strike Beacon Collection

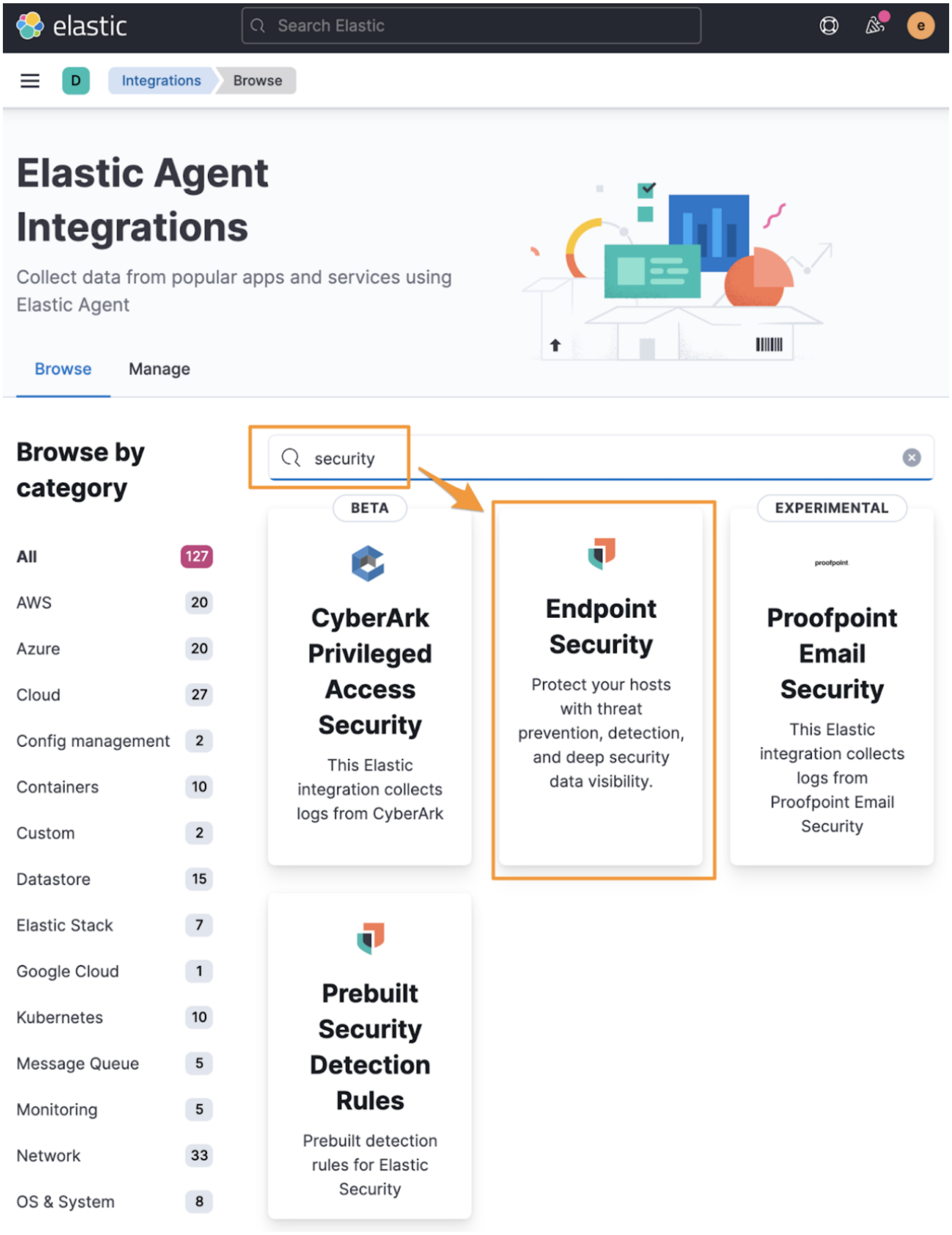
This will be used to collect Cobalt Strike beacons.

Revision	Integrations	Used by	Last updated on
3	0	0 agents	Nov 23, 2021

Add your first integration

This policy does not have any integrations yet.

In the search box, type “security” and select “Endpoint Security”.



Elastic Agent Integrations

Collect data from popular apps and services using Elastic Agent

Browse by category

All	127
AWS	20
Azure	20
Cloud	27
Config management	2
Containers	10
Custom	2
Datastore	15
Elastic Stack	7
Google Cloud	1
Kubernetes	10
Message Queue	5
Monitoring	5
Network	33
OS & System	8

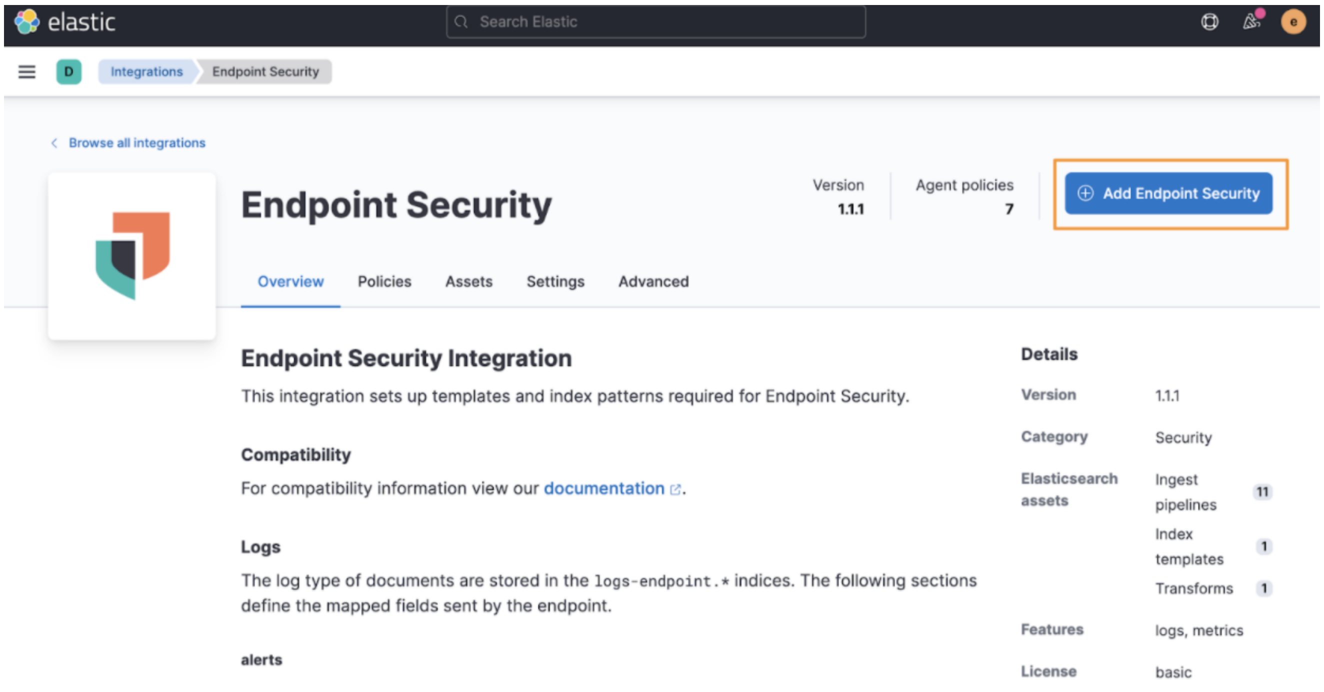
BETA
CyberArk Privileged Access Security
This Elastic integration collects logs from CyberArk

Endpoint Security
Protect your hosts with threat prevention, detection, and deep security data visibility.

EXPERIMENTAL
Proofpoint Email Security
This Elastic integration collects logs from Proofpoint Email Security

Prebuilt Security Detection Rules
Prebuilt detection rules for Elastic Security

Next, click the “Add Endpoint Security” button.



Endpoint Security

Version	Agent policies
1.1.1	7

Endpoint Security Integration

This integration sets up templates and index patterns required for Endpoint Security.

Compatibility

For compatibility information view our documentation.

Logs

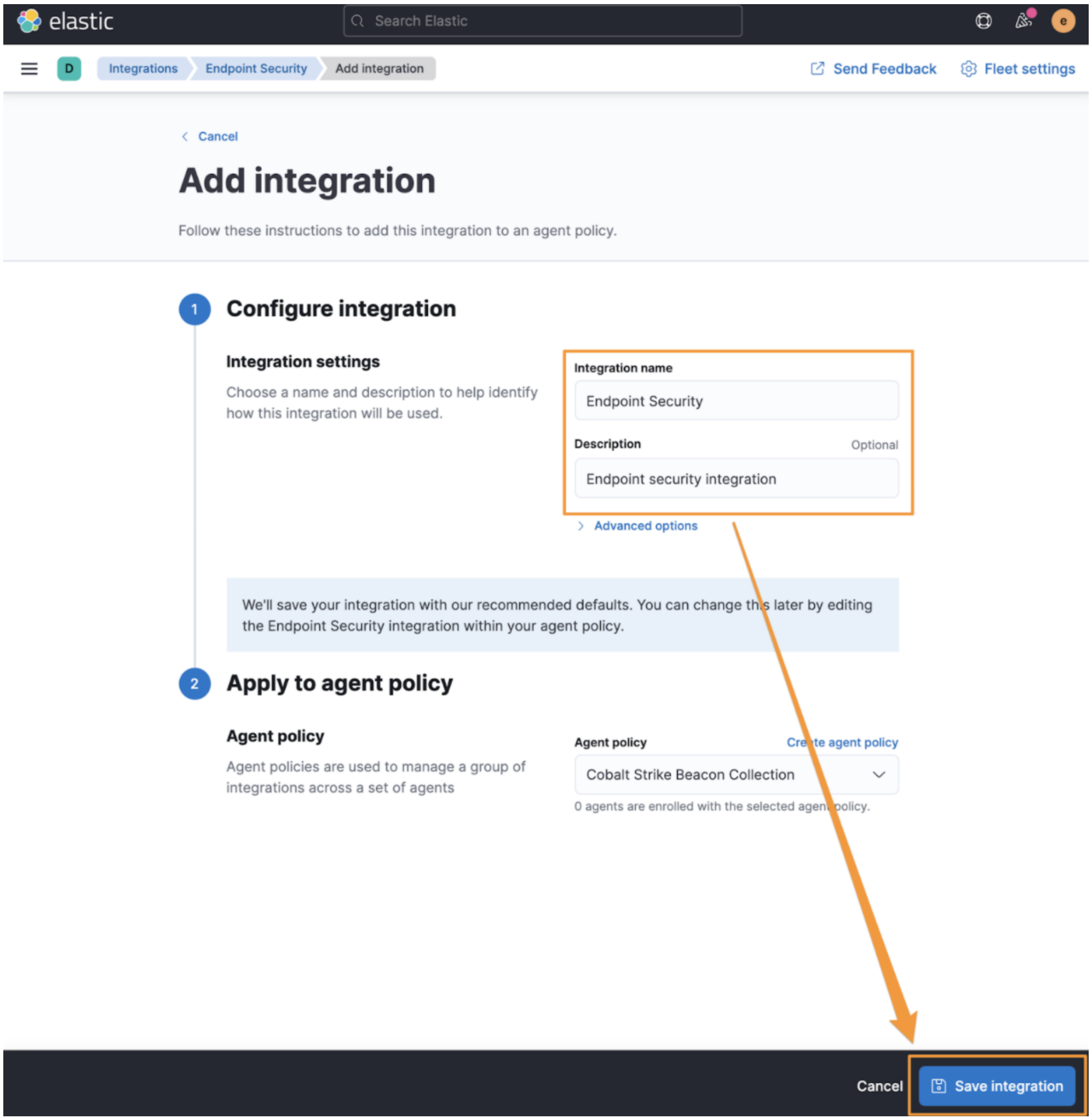
The log type of documents are stored in the `logs-endpoint.*` indices. The following sections define the mapped fields sent by the endpoint.

alerts

Details

Version	1.1.1	
Category	Security	
Elasticsearch assets	Ingest pipelines	11
	Index templates	1
	Transforms	1
Features	logs, metrics	
License	basic	

Next, you'll give the integration a name and description, and then click "Save integration".



Add integration

Follow these instructions to add this integration to an agent policy.

1 Configure integration

Integration settings

Choose a name and description to help identify how this integration will be used.

Integration name: Endpoint Security

Description (Optional): Endpoint security integration

We'll save your integration with our recommended defaults. You can change this later by editing the Endpoint Security integration within your agent policy.

2 Apply to agent policy

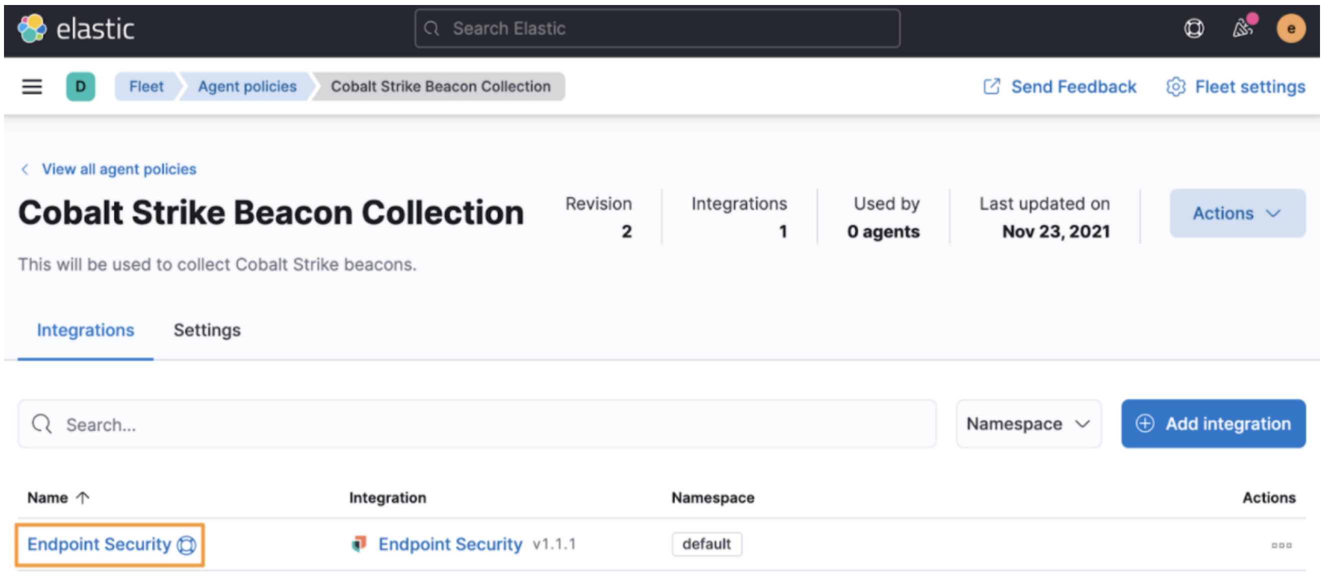
Agent policy

Agent policies are used to manage a group of integrations across a set of agents

Agent policy: Cobalt Strike Beacon Collection

0 agents are enrolled with the selected agent policy.

Finally, we're going to add the memory and shellcode collection options. Click on the integration name ("Endpoint Security").



Cobalt Strike Beacon Collection

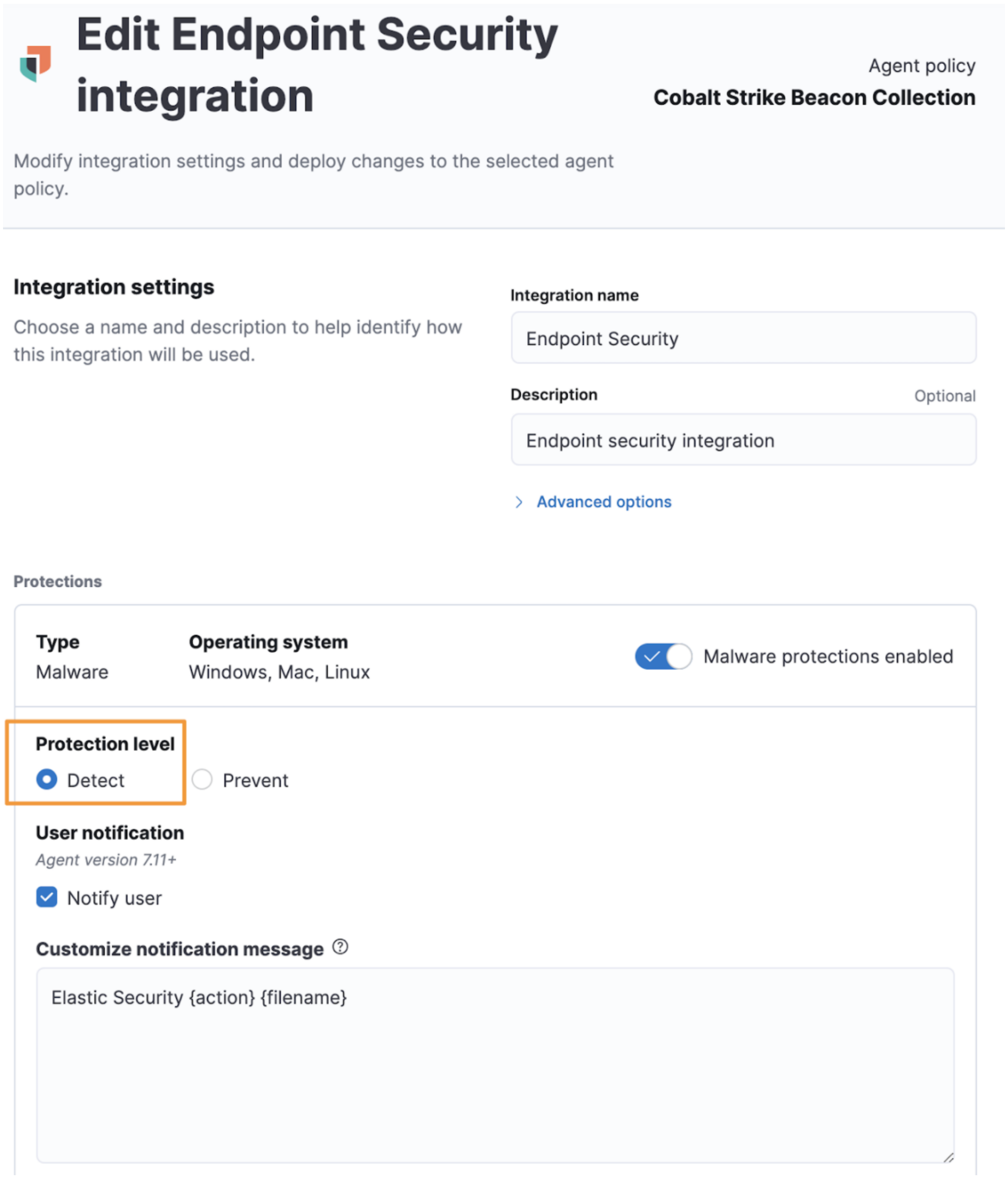
Revision 2 | Integrations 1 | Used by 0 agents | Last updated on Nov 23, 2021

Actions

This will be used to collect Cobalt Strike beacons.

Name	Integration	Namespace	Actions
Endpoint Security	Endpoint Security v1.1.1	default	···

Under “Protections”, leave the different protection types selected, but change the Protection level from “Prevent” to “Detect”. This will allow malware to continue to run to allow for more rich event collection. There are several types of Protections (Malware, Memory, etc.), select “Detect” for each type that has Windows as an available “Operating system”; you can uncheck Mac and Linux Operating Systems. **If you are enabling this feature for a production environment, leave the Protection levels as “Prevent”**



Edit Endpoint Security integration

Agent policy
Cobalt Strike Beacon Collection

Modify integration settings and deploy changes to the selected agent policy.

Integration settings

Choose a name and description to help identify how this integration will be used.

Integration name: Endpoint Security

Description (Optional): Endpoint security integration

Protections

Type	Operating system
Malware	Windows, Mac, Linux

✓ Malware protections enabled

Protection level: [✓ Detect] Prevent

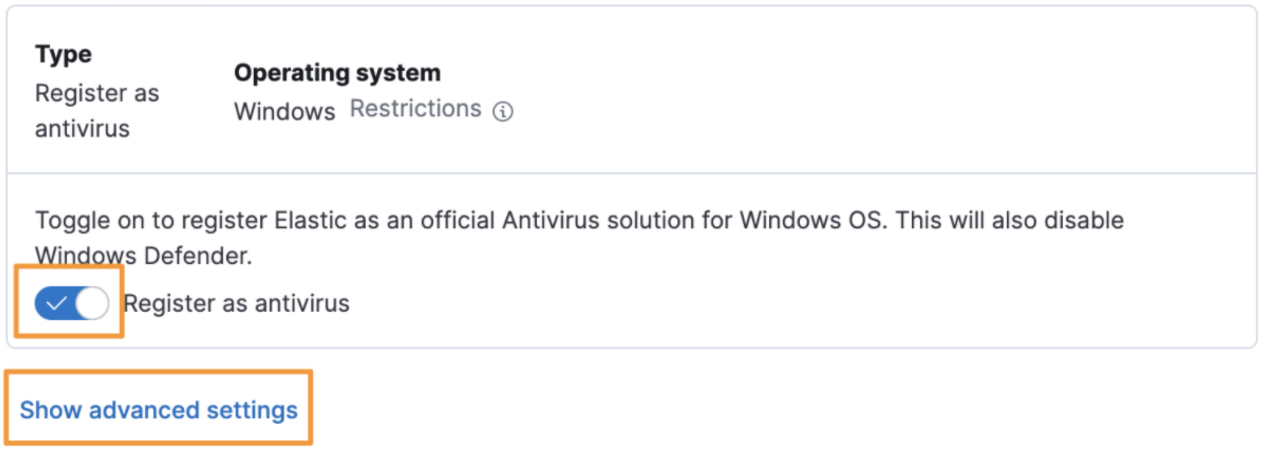
User notification
Agent version 7.11+

✓ Notify user

Customize notification message:

```
Elastic Security {action} {filename}
```

At the bottom of the integration configuration page, you can toggle “Register as antivirus” so that the Elastic Agent is registered as the Antivirus solution, and disable Windows Defender. Click on “Show advanced settings”.

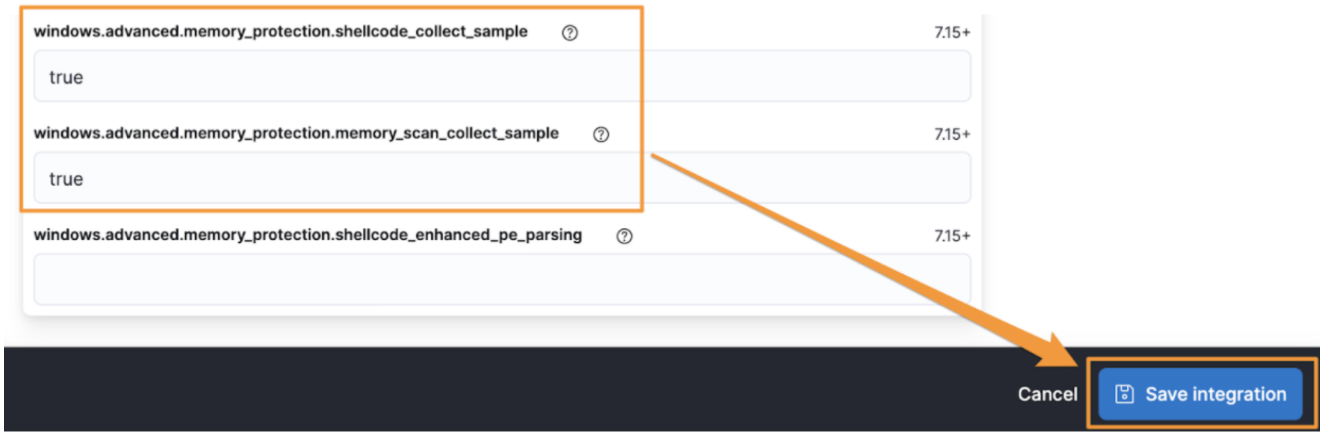

Type
Register as antivirus

Operating system
Windows Restrictions

Toggle on to register Elastic as an official Antivirus solution for Windows OS. This will also disable Windows Defender.

Register as antivirus

At the very bottom of the advanced settings page, type “true” for the `windows.advanced.memory_protection.shellcode_collect_sample` and `windows.advanced.memory_protection.memory_scan_collect_sample` settings, and then click “Save integration”.


windows.advanced.memory_protection.shellcode_collect_sample 7.15+
true
windows.advanced.memory_protection.memory_scan_collect_sample 7.15+
true
windows.advanced.memory_protection.shellcode_enhanced_pe_parsing 7.15+

Once you have created this specific Fleet policy, you can apply this policy to an endpoint running the Elastic Agent. For specific instructions on how to deploy the Elastic Agent, refer to the official Elastic documentation.

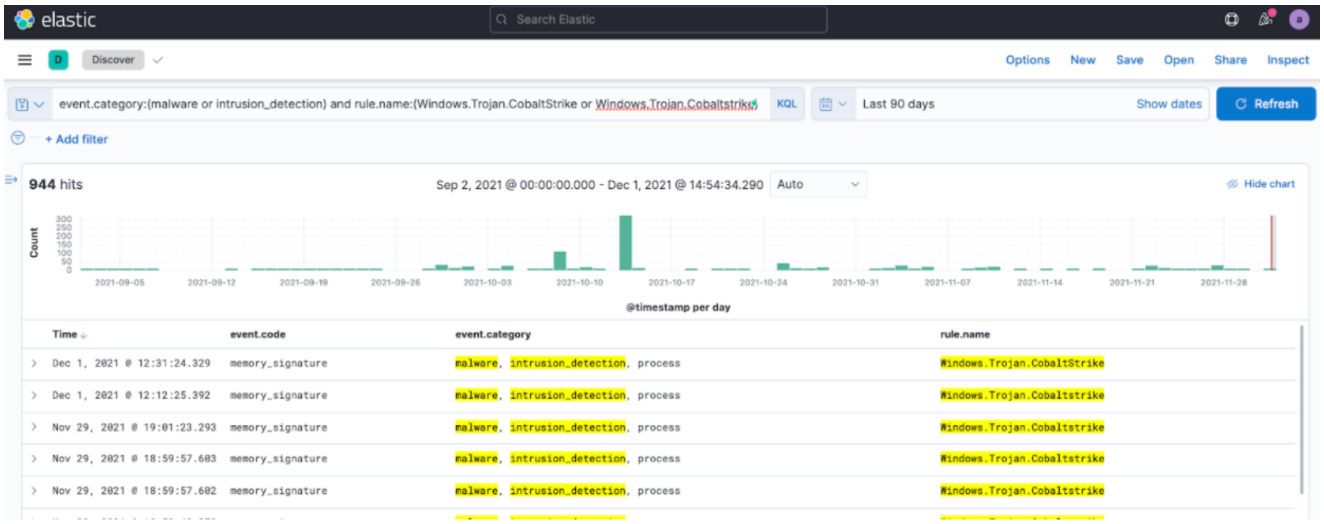
Collecting the Beacon¶

Now that we’ve made a collection policy and applied it to a Windows machine you can target it with a CobaltStrike campaign. Instead of mimicking what a CobaltStrike beacon could look like in a lab, we’re going to use live CobaltStrike beacon payloads from Elastic’s telemetry.

To find Cobalt Strike beacon payloads, you can use the Discover app in Kibana to return events identified as Cobalt Strike. These events are provided by the Elastic Endpoint Security Agent, which identifies Cobalt Strike beacons and modules with the “Windows.Trojan.CobaltStrike” malware signature. A simple Kibana Query Language (KQL) search is as simple as:

KQL search for Cobalt Strike

```
event.category:(malware or intrusion_detection) and
rule.name:(Windows.Trojan.CobaltStrike or Windows.Trojan.Cobaltstrike)
```



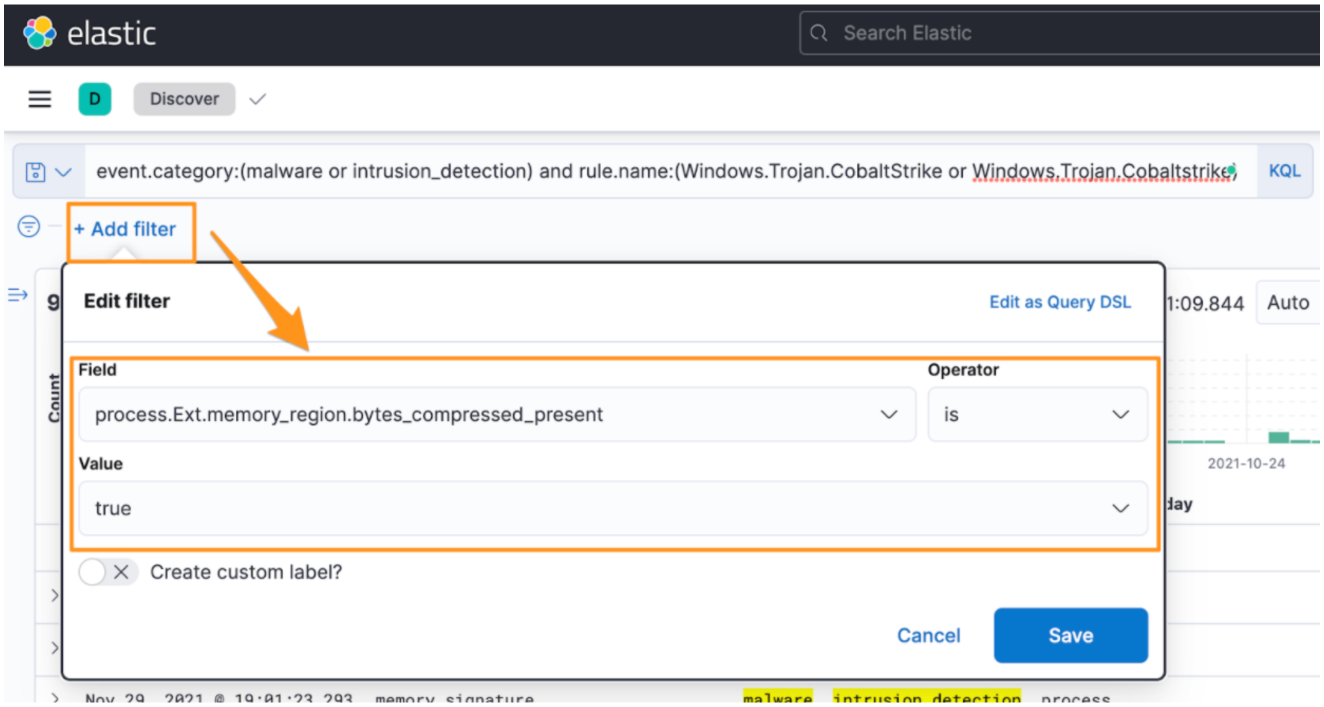
event.category:(malware or intrusion_detection) and rule.name:(Windows.Trojan.CobaltStrike or Windows.Trojan.Cobaltstrike)

944 hits

Sep 2, 2021 @ 00:00:00.000 - Dec 1, 2021 @ 14:54:34.290 Auto

Time	event.code	event.category	rule.name
Dec 1, 2021 @ 12:31:24.329	memory_signature	malware, intrusion_detection, process	Windows.Trojan.CobaltStrike
Dec 1, 2021 @ 12:12:25.392	memory_signature	malware, intrusion_detection, process	Windows.Trojan.Cobaltstrike
Nov 29, 2021 @ 19:01:23.293	memory_signature	malware, intrusion_detection, process	Windows.Trojan.Cobaltstrike
Nov 29, 2021 @ 18:59:57.603	memory_signature	malware, intrusion_detection, process	Windows.Trojan.Cobaltstrike
Nov 29, 2021 @ 18:59:57.602	memory_signature	malware, intrusion_detection, process	Windows.Trojan.Cobaltstrike

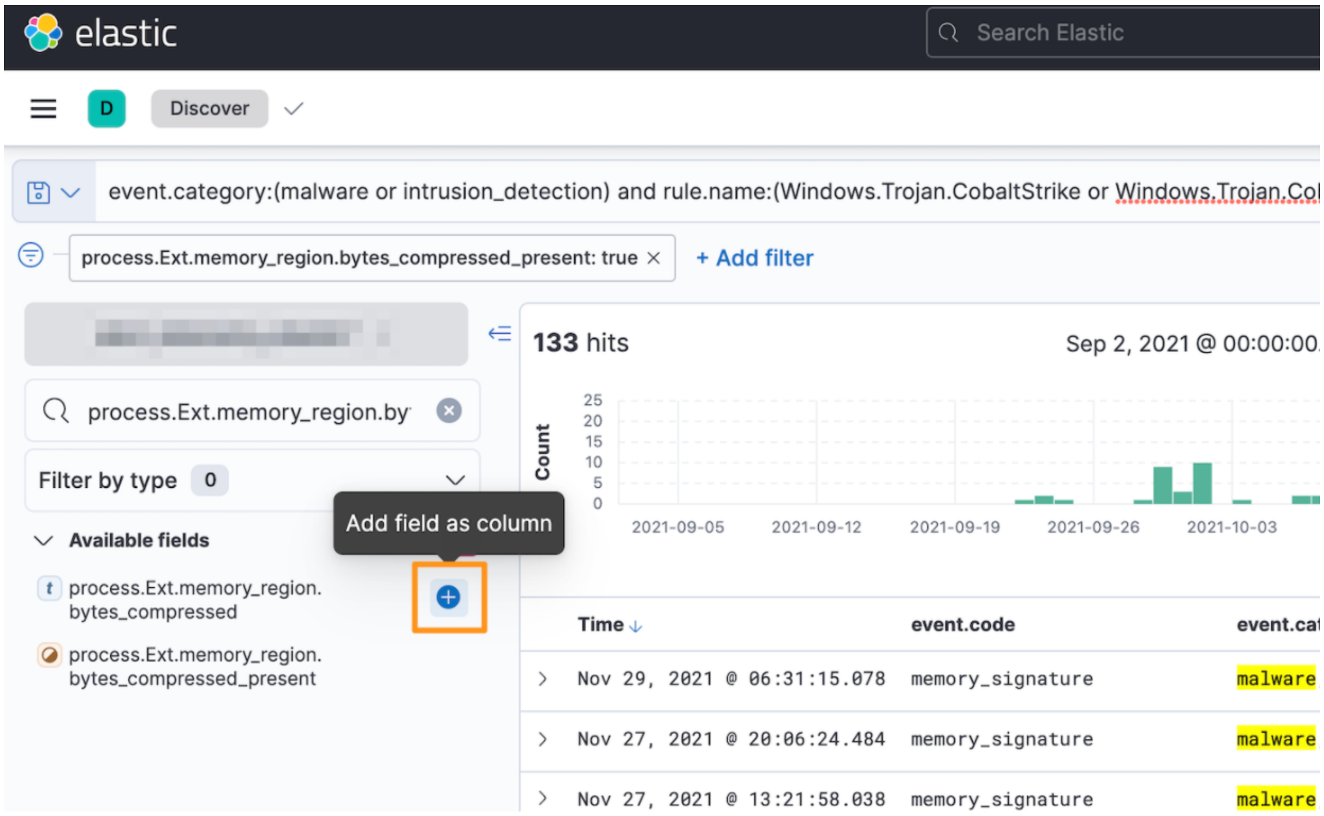
Next, let's filter on documents that have the `process.Ext.memory_region.bytes_compressed` field (this is a field populated by the `windows.advanced.memory_protection.shellcode_collect_sample` and `windows.advanced.memory_protection.memory_scan_collect_sample` settings we configured in the Fleet policy above). To do that we can simply add a filter for the `process.Ext.memory_region.bytes_compressed_present` field with a value of `true`.



event.category:(malware or intrusion_detection) and rule.name:(Windows.Trojan.CobaltStrike or Windows.Trojan.Cobaltstrike

Field: process.Ext.memory_region.bytes_compressed_present
Operator: is
Value: true

Finally, add the `process.Ext.memory_region.bytes_compressed` field to our view so that we can see the value of the field.



event.category:(malware or intrusion_detection) and rule.name:(Windows.Trojan.CobaltStrike or Windows.Trojan.Col

process.Ext.memory_region.bytes_compressed_present: true

process.Ext.memory_region.by

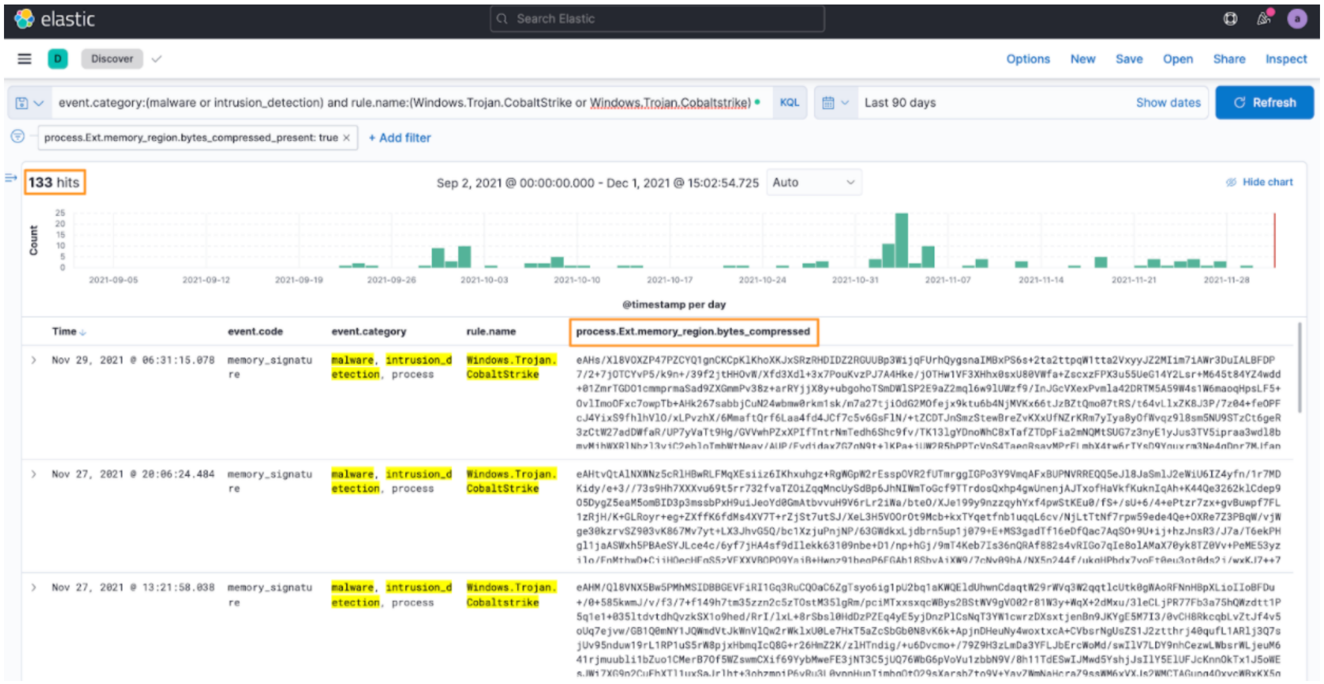
process.Ext.memory_region.bytes_compressed

process.Ext.memory_region.bytes_compressed_present

133 hits Sep 2, 2021 @ 00:00:00.

Time	event.code	event.cat
Nov 29, 2021 @ 06:31:15.078	memory_signature	malware
Nov 27, 2021 @ 20:06:24.484	memory_signature	malware
Nov 27, 2021 @ 13:21:58.038	memory_signature	malware

We can see that we have 133 examples with data in the `process.Ext.memory_region.bytes_compressed` field. This field contains the file extracted from the memory of the infected host and then zlib deflated and Base64 encoded..



event.category:(malware or intrusion_detection) and rule.name:(Windows.Trojan.CobaltStrike or Windows.Trojan.Cobaltstrike)

Last 90 days

process.Ext.memory_region.bytes_compressed_present: true

133 hits

Sep 2, 2021 @ 00:00:00.000 - Dec 1, 2021 @ 15:02:54.725

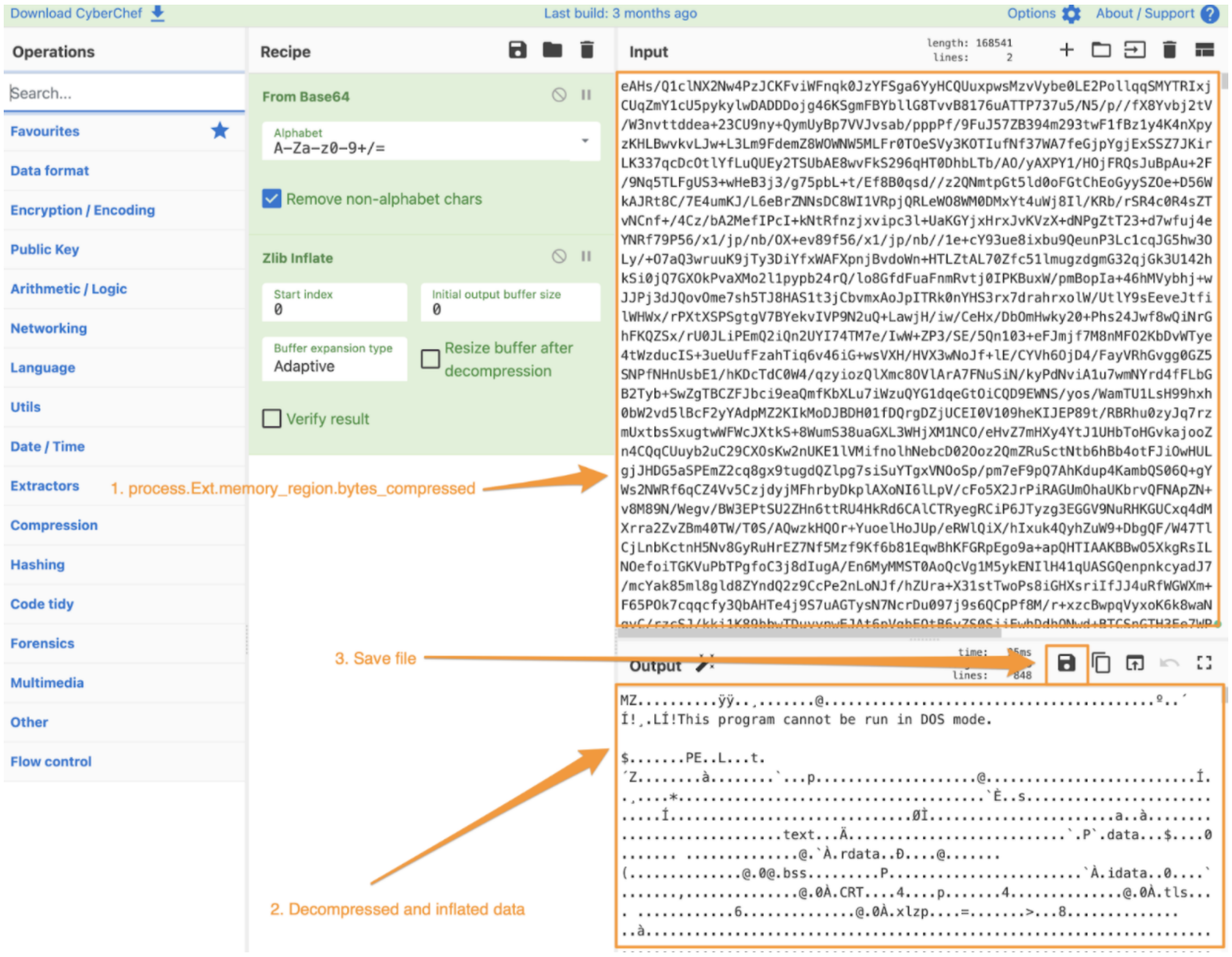
Time	event.code	event.category	rule.name	process.Ext.memory_region.bytes_compressed
Nov 29, 2021 @ 06:31:15.078	memory_signature	malware, intrusion_detection, process	Windows.Trojan.CobaltStrike	eAHs/Xl8VOXZP47PZCYQ1gnCKCpKlKhoXKJxSRzRHDIDZ2RGUUBp3WijqFUrhQygsnaIMBxPS6s+2ta2ttpqW1tta2VxyyJZ2MIim7iAWr3DuIALBFDP7/2+7jOTCYvP5/k9n+/39f2jtHHOvW/Xfd3Xdl+3x7PouKvzPJ7A4Hke/jOTHw1VF3XHhx0sxU80VWfa+ZscxzFPX3u55UeG14Y2Lsr+M645t84YZ4wdd+01ZmrTGDO1cmmprmaSad9ZXGmmPv38z+arRYjjX8y+ubgohoTSmDWlSP2E9aZ2mql6w9lUWzf9/InJGcVXexPvmla42DRTM5A59W4s1W6maoqHpsLF5+OvlImoOFxc7owpTb+AHk267sabbjCuN24wbmw0rkm1sk/m7a27tjiOdG2MOfejx9ktu6b4NjMVKx66tJzBZtQmo07tRS/t64vLlxZK8J3P/7z04+feOPFcJ4YixS9fhlhVlO/xLPvzhX/6MmaftQrf6Laa4fd4JCf7c5v6GsFlN/+tZCDTJnSmzStewBreZvKXxUfNZrKRm7yIya8yOfWvqz918sm5NU9STzCt6geR3zCtW27adDWfaR/UP7yVaTt9Hg/GVVwhPZxXPIfTntrNmTedh6Shc9fv/TK13lgYDnoWhC8xTafZTDpFia2mNQMtSUG7z3nyE1yJus3TV5ipraa3wdl8bmvMihWXRlNhzl3viC2ehloTmhWtNeav/AUP/Fvdidax7G7oN9t+lKPa+iUW2R5hPPTcVnS4TaeoRsavMPrFLmhX4tw6rTYsD9Youxrm3Ne4gDnr7MJfan
Nov 27, 2021 @ 20:06:24.484	memory_signature	malware, intrusion_detection, process	Windows.Trojan.CobaltStrike	eAHtvQtAlNXWNz5cRlHBwRLFMqXEsiiz6IKhxuhgz+RgWGpW2rEsspOVR2fUTmrggIGPo3Y9VmqAFxBUPNVRREQQ5eJl8JaSmlJ2eWiU6IZ4yfn/1r7MDKidy/e+3//73s9Hh7XXXvu69t5rr732fvaTZOiZqqMncUySdBp6JhNIWmToGcf9TTrdosQxhp4gwUnenjAJTxofHaVkfKuknIqAh+K44Qe3262klCdep9O5DygZ5eaM5omBID3p3mssbPxH9uiJeoYd0GmAtbvvuH9V6rLr2iWa/bteO/XJe199y9nzzqyhYxf4pwStKEu0/fS+/sU+6/4+ePtzr7zx+gvBuwpf7FL1zRjH/K+GLRoyr+eg+ZXffK6fdMs4XV7T+rZjSt7utSJ/XeL3H5VOOrOt9Mcb+kxTYqetfnb1uqqL6cv/NjLtTtNf7rpw59ede4Qe+OXRe7Z3PBqW/vjWge30kzrvSZ903vK867Mv7yt+LX3JhvG5Q/bc1XzjuPnjNP/63GWdkxLjdbrn5up1j079+E+MS3gadTf16eDfQac7AqSO+9U+ij+hzJnsR3/J7a/T6ekPHgl1jaASWxh5PBAeSYJLce4c/6yf7jHA4sf9dIlekk63109nbe+D1/np+hGj/9mT4Keb7Is36nQRAf882s4vRIGo7qIe8olAMaX70yk8TZ0Vv+PeME53yzilo/FnMthwD+CjjH0ecHFoS5zVFXXVBOPO9YajB+Hwnz91beoP6FGAh18SbvAjXW9/7cNv09bA/NX5n244f/ukoHPhdx7voFt0eu3ot0ds2j/wxKJ7++7
Nov 27, 2021 @ 13:21:58.038	memory_signature	malware, intrusion_detection, process	Windows.Trojan.Cobaltstrike	eAHM/Ql8VNX5Bw5PMhMSIDBBGEVFiRI1Gq3RuCQOaC6ZgTsyo6ig1pU2bq1aKWQEldUhwnCdaqtW29rWVq3W2qqtlcUtk0gWAoRFNnHBpXLioIIoBFDu+/0+585kwmJ/v/f3/7+f149h7tm35zzn2c5zTOstM35lgRm/pciMTxxsxqcWBys2BStWV9gVO02r81W3y+WqX+2dMxu/3leCLjPR77Fb3a75hQWzdtt1P5q1e1+035ltdvtdhQvzkSX1o9hed/RrI/lxL+8rSbsl0HdDzPZEq4yE5yjDnzPlCsNqT3YW1cwrzDXsxtjenBn9JKYgE5M7I3/0vCH8RkcqbLvZtJf4v5oUq7ejvw/GB1Q0mNY1JQWmdVtJkWnVlQw2rWklxU0Le7HxT5aZcSbGb0N8vK6k+ApjnDHeuNy4woxtxcA+CVbsrNgUsZS1J2ztthrj40qufL1ARlj3Q7sjUv95nduw19rL1RP1uS5rW8pjxHbmqIcQ8G+r26HmZ2K/zlHTndig/+u6Dvcmo+/79Z9H3zLmDa3YFLJbErcWoMd/swIlV7LDY9nhCezwLWbsrWLjeuM641rjmuubli1bZuo1CMerB7Of5WZswmCXif69YybMweFE3jNT3C5jUQ76WbG6pVoVu1zbbN9V/8h11TdESwIJMwd5YshjJsIlY5ElUFJcKnnOkTx1J5oWEsJWi7XG9n2CuFbXTl1uxSaJrlht+3ohzmpiP6vRu3L0vppHupTimbqOtO29sXarsh7to9V+Yav7WmNaHcra79ssWM6xVXJs2WMCTAGupq4OxvcWBxKX5q

Now that we've collected the file in the Elastic Stack, let's turn that raw data into a file that we can analyze.

There is a lot of nuance between operating systems on how to decode Base64 and inflate zlib deflated files. If you'd prefer to use your command line or local tools, feel free to do so. That said, CyberChef is a browser-based data parser that is provided for free by the United Kingdom's Government Communications Headquarters (GCHQ).

Using the CyberChef web application, add the "From Base64" and "Zlib Inflate" recipes.

Click on the disk icon to download the inflated binary.



From Base64
Alphabet: A-Za-z0-9+/=
[✓] Remove non-alphabet chars

Zlib Inflate
Start index: 0
Initial output buffer size: 0
Buffer expansion type: Adaptive
[ ] Resize buffer after decompression
[ ] Verify result

length: 168541
lines: 2

```
eAHs/Q1clNX2Nw4PzJCKFviWFnqk0JzYFSga6YyHCQUuxpwsMzvVybe0LE2PollqqSMYTRIxj
CUqZmY1cU5pykylwDADDDojg46KSgmFBYbllG8TvvB8176uATTP737u5/N5/p//fX8Yvbj2tV
/W3nvttddea+23CU9ny+QymUyBp7VVJvsab/pppPf/9FuJ57ZB394m293twF1fBz1y4K4nXpy
zKHLBwvkvLJw+L3Lm9FdemZ8WOWNW5MLFr0TOeSVy3KOTIufNf37WA7feGjpYgjExSSZ7JKir
LK337qcDcOtlYfLuQUEy2TSUbAE8wvFkS296qHT0DhbLTb/AO/yAXPY1/HOjFRQsJuBpAu+2F
/9Nq5TLFgUS3+wHeB3j3/g75pbL+t/Ef8B0qsd//z2QNmtpGt5ld0oFGtChEoGyySZOe+D56W
kAJRt8C/7E4umKJ/L6eBrZNNsDC8WI1VRpjQRLeWO8WM0DMxYt4uWj8Il/KRb/rSR4c0R4sZT
vNCnf+/4Cz/bA2MefIPcI+kNtRfnzjxvipc3l+UaKGYjxHrxJvKVzX+dNPgZtT23+d7wfuj4e
YNRf79P56/x1/jp/nb/OX+ev89f56/x1/jp/nb//1e+cY93ue8ixbu9QeunP3Lc1cqJG5hw3O
Ly/+O7aQ3wruuK9jTy3DiYfxWAFXpnjBvdoWn+HTLZtAL70Zfc51lmugzdgmG32qjGk3U142h
kSi0jQ7GXOkPvaXMo2l1pypb24rQ/lo8GfdFuaFnmRvtj0IPKBuxW/pmBopIa+46hMVybhj+w
JJPj3dJQovOme7sh5TJ8HAS1t3jCbvmxAoJpITRk0nYHS3rx7drahrxolW/UtlY9sEeveJtfi
lWHWx/rPXtXSPSgtgV7BYekvIVP9N2uQ+LawjH/iw/CeHx/DbOmHwky20+Phs24Jwf8wQiNrG
hFKQZSx/rU0JLiPEmQ2iQn2UYI74TM7e/IwW+ZP3/SE/5Qn103+eFJmjf7M8nMFO2KbDvWTye
4tWzducIS+3ueUufFzahTiq6v46iG+wsVXH/HVX3wNoJf+lE/CYVh6OjD4/FayVRhGvgg0GZ5
SNPfNHnUsbE1/hKDcTdC0W4/qzyiozQlXmc8OVlArA7FNuSiN/kyPdNviA1u7wmNYrd4fFLbG
B2Tyb+SwZgTBCZFJbci9eaQmfKbXLu7iWzuQYG1dqeGtOiCQD9EWNS/yos/WamTU1LsH99hxh
0bW2vd5lBcF2yYAdpMZ2KIkMoDJBDH01fDQrgDZjUCEI0V109heKIJEP89t/RBRhu0zyJq7rz
mUxtbsSxugtwWFWcJXtkS+8WumS38uaGXL3WHjXM1NCO/eHvZ7mHXy4YtJ1UHbToHGvkajooZ
n4CQqCUuyb2uC29CXOsKw2nUKE1lVMifnolhNebcD02Ooz2QmZRuSctNtb6hBb4otFJiOwHUL
gjJHDG5aSPEmZ2cq8gx9tugdQZlpg7siSuYTgxVNOoSp/pm7eF9pQ7AhKdup4KambQS06Q+gY
Ws2NWRf6qCZ4Vv5CzjdyjMFhrbyDkplAXoNI6lLpV/cFo5X2JrPiRAGUmOhaUKbrvQFNApZN+
v8M89N/Wegv/BW3EPtSU2ZHn6ttRU4HkRd6CAlCTRyegRCiP6JTyzg3EGGV9NuRHKGUCxq4dM
Xrra2ZvZBm40TW/T0S/AQwzkHQOr+YuoelHoJUp/eRWlQiX/hIxuk4QyhZuW9+DbgQF/W47Tl
CjLnbKctnH5Nv8GyRuHrEZ7Nf5Mzf9Kf6b81EqwBhKFGRpEgo9a+apQHTIAAKBBwO5XkgRsIL
NOefoiTGKVuPbTPgfoC3j8dIugA/En6MyMMST0AoQcVg1M5ykENIlH41qUASGQenpnkcyadJ7
/mcYak85ml8gld8ZYndQ2z9CcPe2nLoNJf/hZUra+X31stTwoPs8iGHXsriIfJJ4uRfWGWXm+
F65POk7cqqcfy3QbAHTe4j9S7uAGTysN7NcrDu097j9s6QCpPf8M/r+xzcBwpqVyxoK6k8waN
```

time: 25ms
lines: 848

```
MZ..........ÿÿ..¸.......@.....................................º..´
Í!¸.LÍ!This program cannot be run in DOS mode.

$.......PE..L...t.
´Z.......à........`...p.....................@..........................Í.
.¸....*.....................................`È..s.........................
.....Í.............................ØÌ.......................a..à........
....................text...Ä...............................`.P`.data...$....0
....... ...............@.`À.rdata..Ð....@.......
(..............@.0@.bss.........P...........................`À.idata..0....`
.......,...............@.0À.CRT....4....p.......4..............@.0À.tls...
. ............6..............@.0À.xlzp....=.......>...8..............
..à........................................................................
```

Running the `file` command, we can see that this is a Portable Executable (PE) file that can be analyzed by a malware reverse engineer (RE).

Using the file command to validate the file type
While an RE can identify a tremendous amount of information, let's explore what additional information a non-RE can obtain from this file.

Next Steps¶

In the next release, we'll use the beacon that we've just collected and extract its configuration. With this information, we'll be able to identify other important elements such as license identifications, watermarks, and atomic indicators.

Last update: January 31, 2022
Created: January 19, 2022