CONTEXT

UrlScan is basically an online service to scan URLs. It is available at urlscan.io. You can submit a URL, and it will query it, indexing every URL the query goes through, every resource fetched, parts of the page content, and some metadata.

What is valuable is that results of public submissions done by other users are accessible to everyone. As many SOC analysts, security researchers, and some sandbox services use UrlScan, it is one of the best services to find infrastructure related to a malicious campaign.

If you are investigating a phishing targeting your company, creating a free account on UrlScan and using it will often prove that you are not the only target of the campaign and will give you tons of intel.

I'm not affiliated in any way with UrlScan, but I just love this service and I want to share some advice on how to find things on it. Be prepared to use your REGEX skills!

USE CASES

- Track phishing infrastructure
- Track Command and Control (C2) infrastructure
- Track Traffic Distribution Systems (TDS) and affiliate marketing scams infrastructure
- Find domains hosting a specific payload
- Find domains hosted on a specific IP

LIMITATIONS

While using UrlScan, you should keep in mind its limitations:

- **Telemetry is not exhaustive**: If you don't find something, it may exist elsewhere. Just hunt harder!
- **Not everything on UrlScan is malicious**: You should verify that things that appear malicious really are
- **Baddies will often try to mess with automatic analysis**: Infection/redirect chains will often be incomplete, and you should be prepared to be rick-rolled to death.

ALTERNATIVES

Similar URL scanning services are hosted at urlcna.io, urljquery.net, Looklyloo, an URL scanning service developed by CERT.LU, is also worth trying. If you need to scan plenty of suspicious websites with a self-hosted tool, I recommend gowitness *(disclaimer: made by unknown people working for my employer)*.

SEARCHES

To use search filters, you should have an account; a free account is definitely worth it. Some queries may require you to be a Pro user and will be indicated with . Full search description is accessible here, but the goal is to share the search techniques I use the most. The syntax to be used is generally `<search-filter>:<value>`. Filters can be combined with the capitalized operators `OR`, `AND`, `NOT`, and can also be grouped with `()`.

Field	Description
`page.domain`, `domain`	Contacted domains
`page.url`	Contacted URLs
`page.ip`, `ip`	Contacted IPs
`page.status`	HTTP Response code
`page.server`	HTTP "Server" header of primary request
`hash`	Hash from a page ressource
`filename`	Any URL that was requested
`stats.datalength`	Data size of all subresources
`contentInputNames`	Name attributes of input fields on page

Field	Description
`contentGlobalNames`	Names of non-standard JavaScript global variables
`textContent`	Visible text on the website, truncated to the first 20kB
`pageTitle`	Title of the page

Remarks: `page`-prefixed fields contain data related to the submitted URL and its redirect chain, while the equivalent field without the prefix also contains data for the URL's resources.

OTHER THINGS TO KNOW

UrlScan Search API is accessible even without an account. It could be useful for fetching all the results of a search in a structured format. The obtained JSON can then be easily filtered with `jq` or other similar tools.

```
# Query is pased URL encoded in the q parameter
curl "https://urlscan.io/api/v1/search/?q=filename%3A%22337946.png%22%20AND%20domain%3A%22bubbleapps.io%22" -o results.json
# Filter JSON
jq -r '.results[] | [.task.uuid, .task.url, .page.url, .page.title] | @csv ' results.json
```

When getting results from the UrlScan API, not all interesting data are contained inside the returned JSON. For example, the DOM content is not present. You can use the `.task.uuid` field to get the DOM content of a result by querying `https://urlscan.io/dom/<UUID>`. This approach can be really handy when scripting data extraction from the DOM of a page, such as outgoing links, advertising IDs, or next-stage payload URLs.

```
jq -r '.results[].task.uuid' results.json | while read uuid; do
  curl --compressed "https://urlscan.io/dom/$uuid" -o "${uuid}.html"
done
```


PIVOTS

Goal	Query
Bulk domain search, find all pages associated to a list of domains	`domain:(domain1 OR domain2 OR ...)`
Search for domains matching a pattern or a TLD	`domain:/outlook.*\.info/`
Search for a specific pattern in URL (here possible Mintsloader C2)	`page.url:/.*1\.php\?s=.*/`

Remarks: Making pivots for phishing under a legitimate domain is a good way to improve your knowledge of UrlScan. It forces you to try many solutions and to find circumventions around the limited amount of data exposed by UrlScan.

Goal	Query	Comment
Find phishing websites hosted under a legitimate website with content	`text.content:" Ihr Dokument wurde erfolgreich" AND domain:"bubbleapps.io"`	Filtering by content allows tracking a campaign for a specific geo. However, you will likely miss some results.
Find phishing websites	`contentInputNames:"company" AND`	Filtering by the `name` attribute of an HTML input
hosted under a legitimate website with forms	`domain:"bubbleapps.io"`	sometimes allows isolating a specific campaign.
Find phishing websites hosted under a legitimate website with resources	`filename:"337946.png" AND domain:"bubbleapps.io"` or `hash:5eedb[...]d309 AND domain:"bubbleapps.io"`	When consulting a result page, in the HTTP tab, you can see filenames and resource hashes. Never forget to click any suspicious ones to try clustering things!
Find phishing websites hosted under a legitimate website with stats	`stats.requests:17 AND domain:"bubbleapps.io"`	Returns some false positives, but it is still better than nothing!