A Growing Mongo Problem

Ming Chow
Email: mchow@cs.tufts.edu
Twitter: @0xMchow
MongoDB Use in the Present

- Massively Multiplayer Online Games (MMORPGs): EA’s FIFA Online 3 (http://www.mongodb.com/blog/post/ea-scores-mongodb-based-fifa-online-3)
- Countless companies including Adobe, Craigslist, Dropbox, eBay, MetLife (http://www.mongodb.com/mongodb-scale)
The Internet of Things (IoT)
- Embedded devices
- For real-time data from sensors, connected devices

References:
Why Target MongoDBs?

- For the *enormous* amount of data stored.
- Naive developers and administrators => they hardly care about security; defaults are almost never changed
  - Learning curve of using MongoDB is low
- 10Gen is oblivious to security issues: "...We were on with...the MongoDB guys talking about the security of the platform, and...it was really clear that they just didn’t care, because their customers weren’t asking for it." - Rich Mogull, Security Weekly Podcast Episode 345
The Reality of MongoDB and Security: It Is That Bad

- “Just walk in”
- No authentication out-of-the-box
- Lots of blind trust
  - User input
  - Connections from other computers
- No encryption used for data or for transmission of data from server to client by default
On SHODAN
On SHODAN (continued)
On Project Un1c0rn (http://un1c0rn.net/)
On Project Un1c0rn (continued)
On the Application Side, Injection Attacks

1. **Query**: creating unsafe queries via string concatenation
2. **Schema**: inserting a record into a schema that does not exist will automatically create the new schema
3. **JavaScript**: `$_where`, `db.eval()` take in JavaScript functions as parameters
4. **Request**: insert associative arrays in GET requests. See https://gist.github.com/mchow01/49f8979829f1c488d922
Russell Butturini's NoSQLMap

- [http://www.nosqlmap.net/](http://www.nosqlmap.net/)
- Automates injection attacks and exploit default configuration
- Supports MongoDB and now also CouchDB
- Notable features on MongoDB:
  - Database enumeration and cloning attacks
  - Extracts database names, users, and password hashes
  - Scans subnets or IP lists for MongoDB databases with default access and enumerating versions
  - Dictionary and brute force password cracking of recovered MongoDB hashes
1-Set target host/IP (Current: 
2-Set web app port (Current: 80)
3-Set App Path (Current: Not Set)
4-Toggle HTTPS (Current: OFF)
5-Set MongoDB Port (Current: 27017)
6-Set HTTP Request Method (GET/POST) (Current: GET)
7-Set my local MongoDB/Shell IP (Current: Not Set)
8-Set shell listener port (Current: Not Set)
9-Toggle Vorbis Mode: (Current: ON)
0-Load options file
a-Load options from saved Burp request
c-Save options file
x-Back to main menu
Select an option: x

NoSQLMap
NoSQLMap-v0.5
nosqlmap@gmail.com

1-Set options
2-NoSQL DB Access Attacks
3-NoSQL Web App attacks
4-Scan for Anonymous MongoDB Access
5-Change Platform (Current: MongoDB)
x-Exit
Select an option: 2
DB Access attacks (MongoDB)
--------------------
Checking to see if credentials are needed...
Successful access with no credentials!
Now What?

- What I underestimated: the flexibility and scaling of MongoDB.
- With the growing concerns of security of the “Internet of Things”, this only adds fuel to the fire.
- The future is ripe for attackers.
References

- Chow, M. “Abusing NoSQL Databases” DEF CON 21, Las Vegas, NV

- Butterini, R. “Making Mongo Cry: NoSQL for Penetration Testers” DerbyCon 2014, Louisville, KY
  http://slideplayer.us/slide/2510942/, Video: