DEFCON XXI

How my Botnet Purchased Millions of Dollars in Cars & Defeated the Russian Hackers
I want to tell you a story about...

- Hacking
- Cars
- Russian Hackers
- Screwing with the system
I want to tell you a story about...

- Commercial Botnets
- Creating competitive advantages
- Not using technology as directed
What you'll learn

• What makes a good Botnet / Webbot project?
• How Bots create competitive advantages for business (example)
• What I would do differently today
  (the example happened 6 years ago)
What makes a good Bot project?

- The Bot...
- Can't recreate Google
- Must solve a problem
- Be viable for it's service life
- Doesn't "show it's hand"
I have permission to tell this story.

- It's rare that I get to mention specific projects.

- I've been writing about bots since 1999
  - Medical diagnostics
  - Privacy
  - Fraud detection
  - Private investigations
  - Governments
So the 1st thing I did, was write this..
The Problem to be Solved

- Dealerships that sell new cars make most of their money on used vehicles.
- Automobile dealers spend a lot of time & money acquiring (previously owned) inventory.
- A client found a website that had great cars for sale.
- Unfortunately, due a lot of competition (and bad web design) he wasn't able to buy the cars he wanted.
The Opportunity

- Daily, a national franchise posted about 300 rental returns for dealerships to purchase.
- They could view the cars before the sale.
- No car could be purchased before 10AM PDT.
### The Opportunity

<table>
<thead>
<tr>
<th>MAKE</th>
<th>Hudson</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODEL</td>
<td>112 Coupe</td>
</tr>
<tr>
<td>YEAR</td>
<td>1938</td>
</tr>
<tr>
<td>MILES</td>
<td>47,000</td>
</tr>
<tr>
<td>CONDITION</td>
<td>Excellent</td>
</tr>
<tr>
<td>PRICE</td>
<td>$18,500</td>
</tr>
</tbody>
</table>

**Price:** $18,500

- **Before 10:00am:**
- **After 10:00am:**

---

For more information or to purchase, visit our website or contact us directly.
<table>
<thead>
<tr>
<th>MAKE</th>
<th>Hudson</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODEL</td>
<td>112 Coupe</td>
</tr>
<tr>
<td>YEAR</td>
<td>1938</td>
</tr>
<tr>
<td>MILES</td>
<td>47,000</td>
</tr>
<tr>
<td>CONDITION</td>
<td>Excellent</td>
</tr>
<tr>
<td>PRICE</td>
<td>$18,500</td>
</tr>
</tbody>
</table>

**Before 10:00am**

<table>
<thead>
<tr>
<th>MAKE</th>
<th>Hudson</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODEL</td>
<td>112 Coupe</td>
</tr>
<tr>
<td>YEAR</td>
<td>1938</td>
</tr>
<tr>
<td>MILES</td>
<td>47,000</td>
</tr>
<tr>
<td>CONDITION</td>
<td>Excellent</td>
</tr>
<tr>
<td>PRICE</td>
<td>$18,500</td>
</tr>
</tbody>
</table>

**After 10:00am**

Users had to continually refresh their browser to get this button to appear.
And here's the real problem.

If there are two hundred available cars...
And here's the real problem.

About **five** cars will be highly desirable
And here's the real problem.

Every franchised dealership...
And here's the real problem.

Attempted to buy the same cars.
What my client faced

- There were a limited number of “real deals”
- Every dealership wanted the same cars
- The website's design created peak bandwidth demands, that made the website very hard to use.
How my client coped

While waiting for this button to appear...

- Car dealers would recruit every available person for this task.
- These people would be assigned to specific cars.
- They'd have multiple browser windows open.
- They'd keep hitting the browser refresh button until it was enabled.
How my client coped

While waiting for this button to appear...

Every franchise (for this brand) did these things

- Car dealers would recruit every available person for this task.
- These people would be assigned to specific cars.
- They’d have multiple browser windows open.
- They’d keep hitting the browser refresh button until it was enabled.
Site design caused a technical problem

- The bandwidth/server lag peaked at the sale time
- It could take as long as 30 seconds for screens to refresh
Site design caused a technical problem.

- The bandwidth/server lag peaked at the sale time.
- It could take as long as 30 seconds for screens to refresh.

The website was basically unusable... if used as designed.
The client called and asked if a webbot could help.
Problems with existing system:

1. Was too manual
2. The “Buy” button took too long to appear
Solution, Part 1

Problems with existing system:
1. Was too manual
2. The “Buy” button took too long to appear

Note:
This was seven years ago, and I don't develop like this anymore.
Solution, Part 1

I developed a light weight web interface like this.
Solution, Part 1

Each client was in an HTML frame.

Initially, we used four instances.

The client would often load the bot on multiple computers.
Solution, Part 1

The first step was to validate the VIN
Once the VIN was validated, it waited for the client to tell it to start.
Solution, Part 1

The Client Bot, then synchronized clocks with the Target
And started the count down to purchase.
Solution, Part 1

Every now and then, we'd miss one.
Solution, Part 1

But more often:

- The sale was successful, and
- The Bot arranged for financing & shipping
The client's purchase success rate went from 0% to near 100%
What accounted for the initial success?

BEFORE

BOT CLIENT

TARGET SERVER

REFRESH

SALE ACTIVE

PURCHASE ATTEMPT

AFTER

BOT CLIENT

TIMER

9 12 6 3

BOT SERVER

PURCHASE ATTEMPT

TARGET SERVER
What accounted for the initial success?

This got slower as time of sale got close.

Before:
- Bot Client
- Target Server
- Purchase Attempt
- Sale Active
- Refresh

After:
- Bot Client
- Bot Server
- Target Server
- Purchase Attempt
- Timer

What accounted for the initial success?
What accounted for the initial success?

The Bot was successful primarily because it didn't have to wait to be told that the sale was on.
A sign of problems

The client continued to have success for about six months.

- Suddenly, success rates dropped to about 50%
- My client discovered a competing bot developed by a group of Russian Hackers
- Competition is good, and leads to innovation.
The clock synchronization was modified to make checks more often as the sale neared, and also calculated system lag.
Solution, Part 2b

Purchases where attempted:

- The Bot Client triggered the Bot Server to make multiple attempts to buy the vehicle.
- Each attempt was made slightly prior to the actual sale time and based on calculated system delays.
How successful was this Bot?

- The Bot operated for about 40 weeks
- The client bought approximately 20 cars a week (estimated)
- Total cars purchased **800** (estimated)
- If the average wholesale cost ~$16,000
- The Bot purchased **$12,800,000** (estimated wholesale value)
What would I do differently?

- What did it do well?
  - More successful than anticipated
  - Very lightweight clients
    - Easily updated
    - Easily distributed
What would I do differently?

- Fairly stealthy
  - It required authenticated users
    - Affects stealthiness
  - Tried to simulate human behavior
  - Using multiple accounts
  - The expected behavior was pretty weird
What would I do differently?

- Today, it would have to accommodate newer technology
  - The sale website used standard HTML forms, which are easy to emulate (submit) with simple PHP scripts
  - Today's websites are more suffocated
    - AJAX
    - Complex forms, etc.
What would I do differently?

- Vehicles could be written into a “task queue” via a web interface to the BOT SERVER

### TASK QUEUE

- VIN: JTHBW1GG3D2012782, STATUS: PENDING
- VIN: JTHBW1GG5D2021449, STATUS: PENDING
- VIN: JTHBW1GG6D2004871, STATUS: PENDING
What would I do differently?

- Tasks in the Task Queue are distributed to individual computers that I call *harvesters.*

The harvesters may be located anywhere:
- Data Center
- Office
- Cloud

<table>
<thead>
<tr>
<th>VIN</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTHBW1GG3D2012782</td>
<td>PENDING</td>
</tr>
<tr>
<td>JTHBW1GG5D2021449</td>
<td>PENDING</td>
</tr>
<tr>
<td>JTHBW1GG6D2004871</td>
<td>PENDING</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
What would I do differently?

- The harvesters create (iMacros) **browser macros** on-the-fly and execute commands directly in Firefox
Once the task is completed, the harvesters communicate back to the Bot Server, which updates the Task Queue.

The results in the Task Queue are communicated to The Client via a web interface.
What would I do differently?

If you're interested in the details of how this is done, look-up my DEFCON 17 talk.

Screen Scraper Tricks: Difficult cases
Thanks to All of You, the DEFCON CFP Goons (particularly Nikita) for allowing me to give my 5th DEFCON talk.

If you like this type of talk, the rest of my talks are on YouTube.

- DEFCON X  Developing Webbots & Agents
- DEFCON XI  Corporate Intelligence
- DEFCON XV  The Incredible Executable Image Exploit
- DEFCON XVII Advanced Scraping: Difficult Cases

(search “schrenk DEFCON”)

mgschrenk@gmail.com  www.schrenk.com