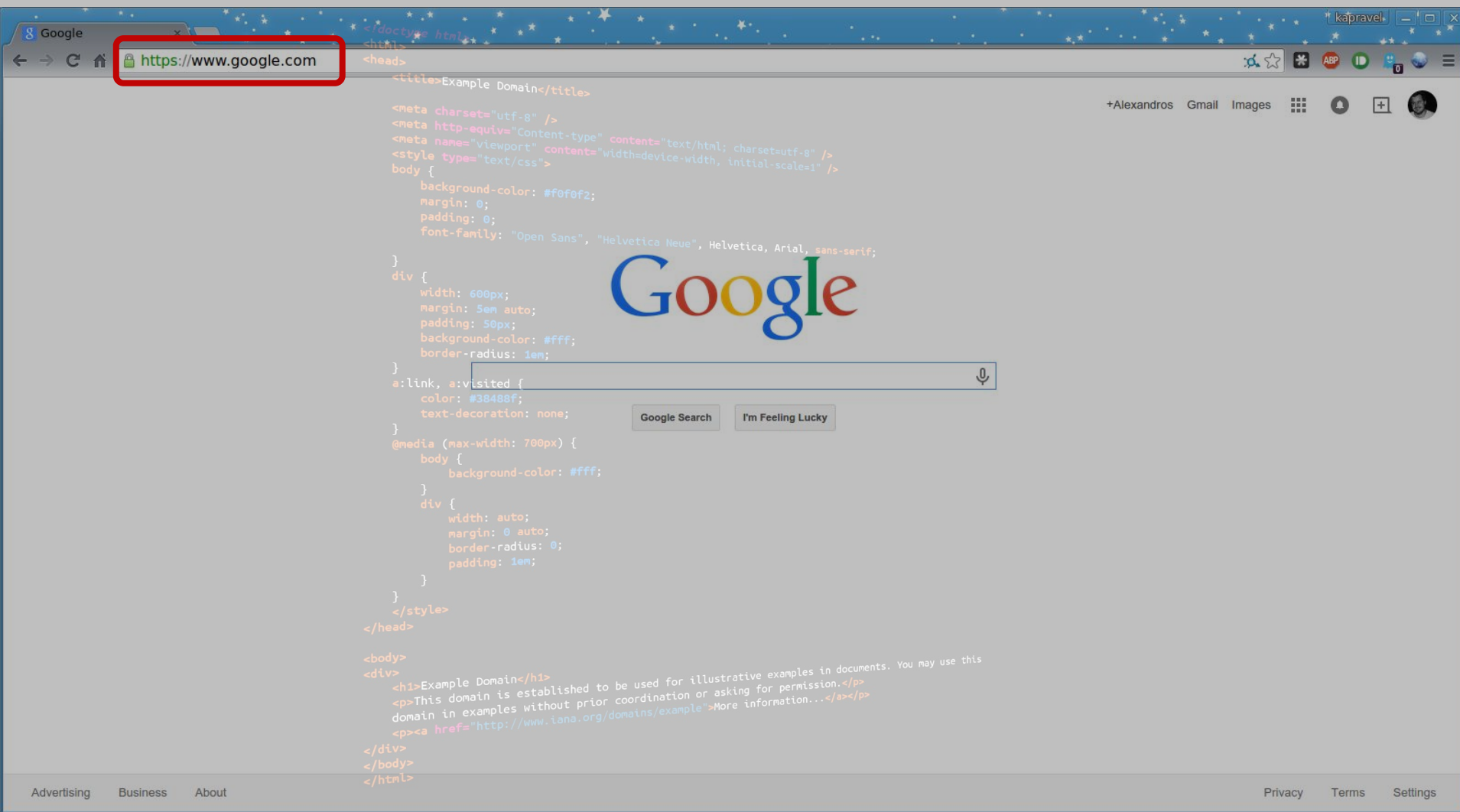


CSC 405

Computer Security

Evasive Web-based Malware

Alexandros Kapravelos
akaprav@ncsu.edu



Giovanni Vigna

cs.ucsb.edu/~vigna/

Search

Giovanni Vigna
Professor

Home Research Teaching Contact

I am a faculty member of the Computer Science Department at the University of California in Santa Barbara.

My research focuses on malware analysis, web security, vulnerability research and intrusion detection.

I am the director of the Center for Cybersecurity at UCSB.

I am co-director of the Security Lab. I am also part of the SocialSecLab.

I am one of the founders of [Securix](#), a company that develops innovative solutions to detect and respond to advanced malware (APMs) and targeted threats.

Every year, I organize the [International Capture The Flag \(CTF\)](#) competition.

Contact Information

Address: Giovanni Vigna
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5120, USA

Phone: (805) 893-3333

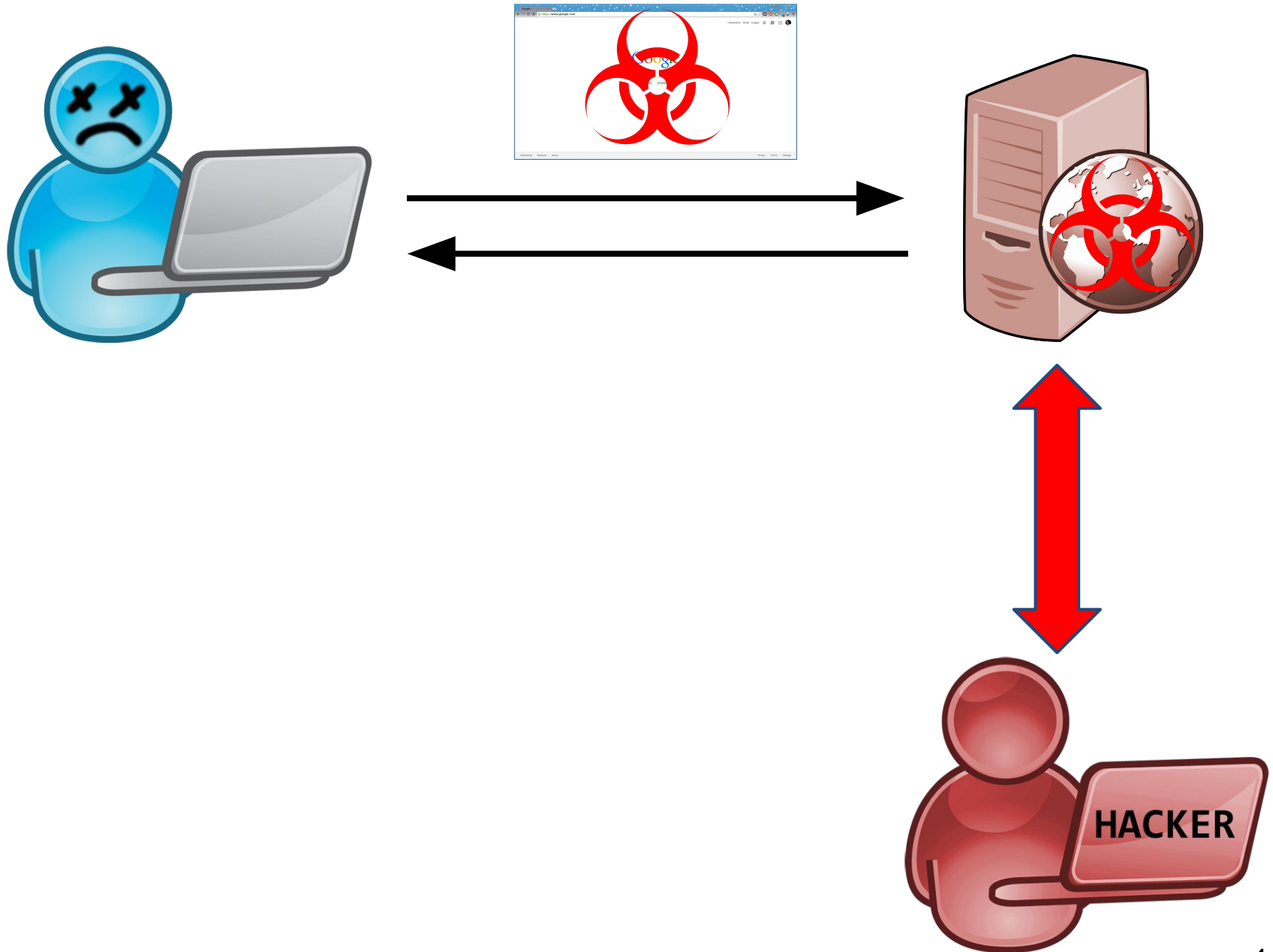
E-Mail: vgigna@ucsb.edu

Web: www.vigna.org

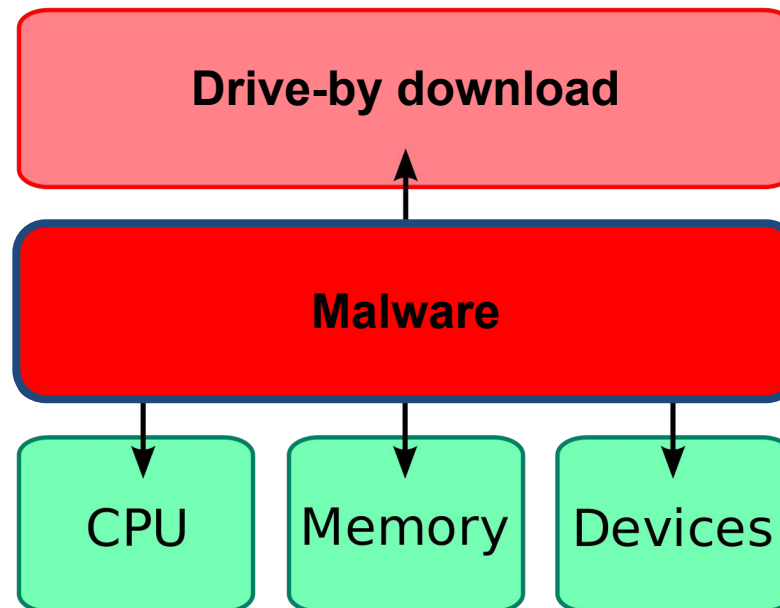
Office: 1010 UCSC Building (1010 UCSC Building) or Room 1010

Blog: blog.vigna.org

Public Key: www.vigna.org/public-key



Compromising the browser





TECHNOLOGY

Google Access Is Disrupted in Vietnam

Some Google users in Vietnam were redirected to a website with the message: 'Hacked by Lizard Squad'



 DATA CENTER SOFTWARE NETWORKS **SECURITY** BUSINESS HARDWARE SCIENCE BOOTNOTES

Rogue ads infiltrate Expedia and Rhapsody

When legit sites attack

Hackers compromise official PHP website, infect visitors with malware (updated)

Php.net goes on lockdown after malicious code is found hosted on site servers.

SECURITY malware

Attack on Dailymotion redirected visitors to exploits

Drive-by download

[illegible]

Latest 0-day exploit - January 2018

Microsoft Edge

33 security fixes

CVE-2018-0758,
CVE-2018-0762,
CVE-2018-0768,
CVE-2018-0769,
CVE-2018-0770,
CVE-2018-0772,
CVE-2018-0773,
CVE-2018-0774,
CVE-2018-0775,
CVE-2018-0776,
CVE-2018-0777, and
CVE-2018-0778

Google Chrome

53 security fixes

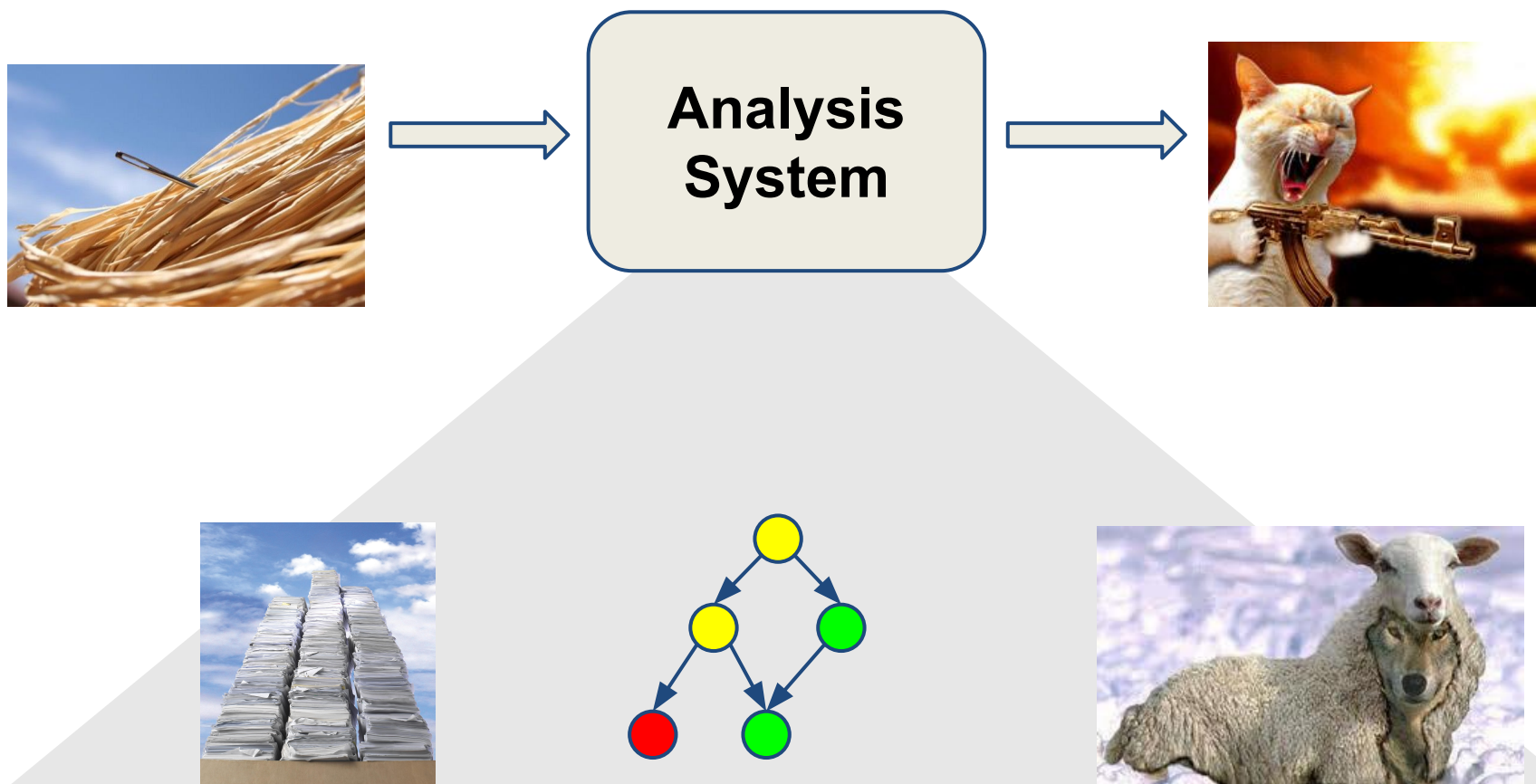
Chrome 64.0.3282.119

Mozilla Firefox

32 security fixes

Firefox 58

Dynamic analysis systems



Wepawet

- System to detect drive-by downloads
- Leading R&D for the past 5 years
- Was publicly available at wepawet.cs.ucsb.edu
- Based on an emulated browser (HtmlUnit+Rhino)
- 93,962,555 processed submissions
- 2,930,669 malicious detections so far
- 1,626 registered users

Wepawet

[Home](#) | [About](#) | [Sample Reports](#) | [Tools](#) | [News](#)

Analysis report for file 90c8f078680a104b4b78810b5a2328ff

Sample Overview

File	variant_72.pdf
MD5	90c8f078680a104b4b78810b5a2328ff
Analysis Started	2015-02-09 16:12:48
Report Generated	2015-02-09 16:14:00
JSAND version	2.3.6

[Reanalyze this file.](#)

Detection results

Detector	Result
JSAND 2.3.6	malicious

malicious

In particular, the following URL was found to contain malicious content:

- file://90c8f078680a104b4b78810b5a2328ff/

Exploits

Name	Description	Reference
Adobe Collab overflow	Multiple Adobe Reader and Acrobat buffer overflows	CVE-2007-5659

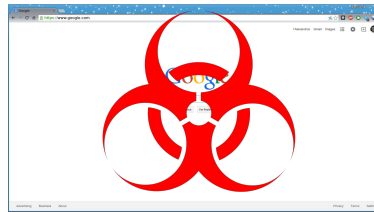
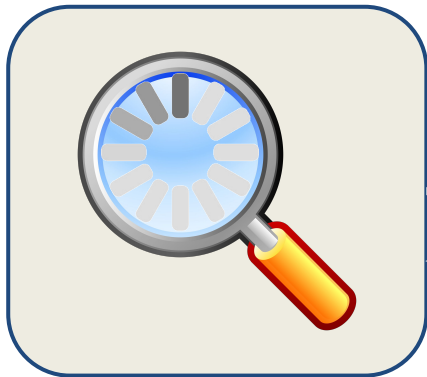
Features

- Redirection and cloaking
- Deobfuscation
- Exploitation

Attack in the wild

JAVASCRIPT

```
var nop="%uyt9yt2yt9yt2";
var nop=(nop.replace(/yt/g,""));
var sc0="%ud5db%uc9c9%u87cd...";
var sc1="%"+ "yutianu" + "ByutianD" + ...;
var sc1=(sc1.replace(/yutian/g,""));
var sc2="%"+ "u" + "54" + "FF" + ... + "8" + "E" + "E";
var sc2=(sc2.replace(/yutian/g,""));
var sc=unescape(nop+sc0+sc1+sc2);
```



evil.com

Wepawet

[Home](#) | [About](#) | [Sample Reports](#) | [Tools](#) | [News](#)

Analysis report for <http://evil.com>

Sample Overview

URL	http://evil.com
Domain	evil.com
Analysis Started	2015-02-03 13:57:19
Report Generated	2015-02-03 17:03:44
JSAND version	2.3.6

[Reanalyze this URL.](#)

[See the report for domain evil.com.](#)

Detection results

Detector	Result
JSAND 2.3.6	benign

Exploits

No exploits were identified.

benign

Evolution from previous sample

JAVASCRIPT

```
try {  
    new ActiveXObject("yutian");  
} catch (e) {  
    var nop="%uyt9yt2yt9yt2";  
    var nop=(nop.replace(/yt/g,""));  
    var sc0="%ud5db%uc9c9%u87cd...";  
    var sc1="%"+ "yutianu" + "ByutianD" + ...;  
    var sc1=(sc1.replace(/yutian/g,""));  
    var sc2="%"+ "u" + "54" + "FF" + ... + "8" + "E" + "E";  
    var sc2=(sc2.replace(/yutian/g,""));  
    var sc=unescape(nop+sc0+sc1+sc2);  
}
```

Detecting the undetected

Revolver

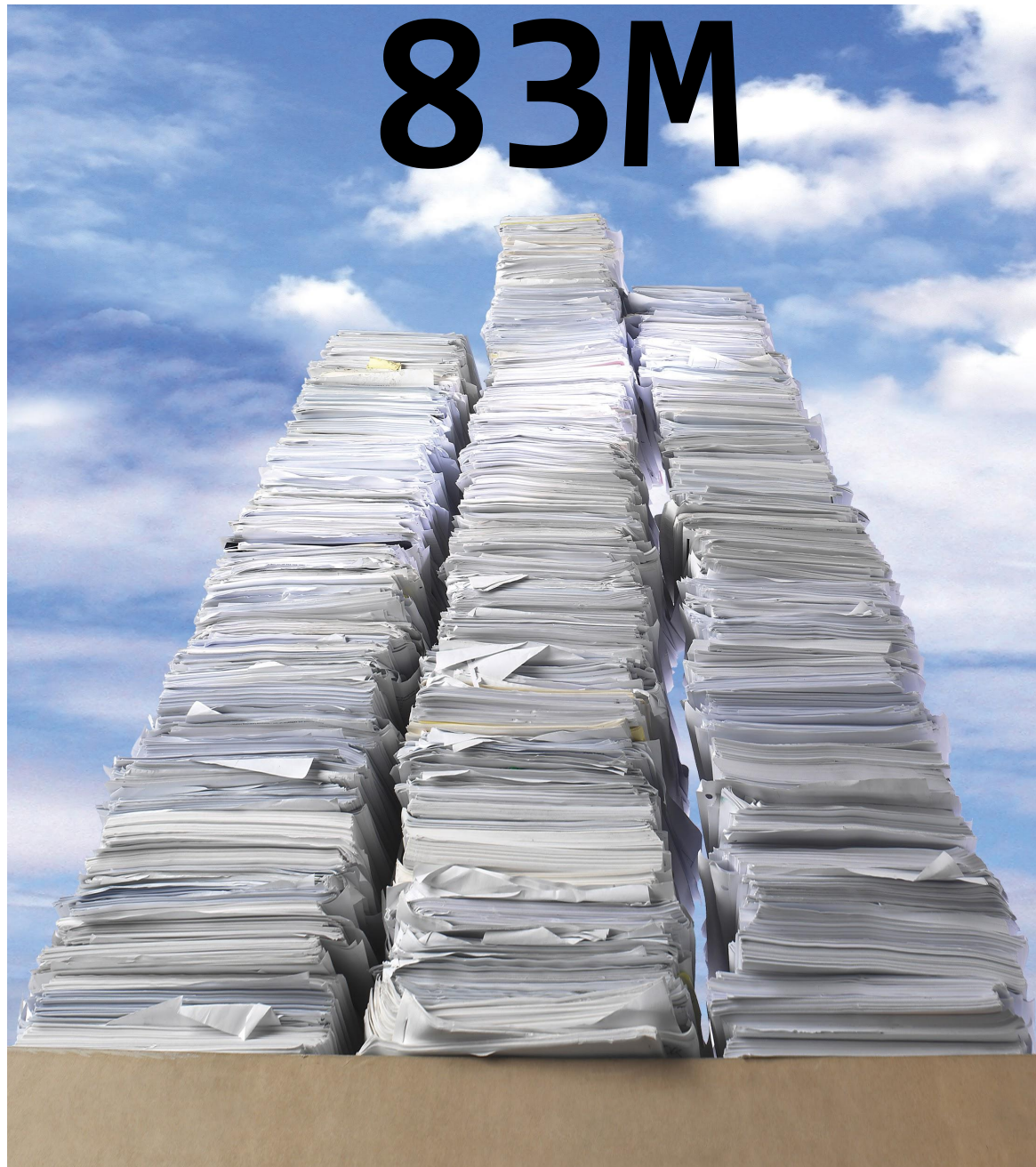


- A system to dynamically track JavaScript evolution
- Was publicly available at revolver.cs.ucsb.edu
- Build on top of Wepawet
- Provides a deep insight into new and previously unseen attacks

Revolver: An Automated Approach to the Detection of Evasive Web-based Malware
Alexandros Kapravelos, Yan Shoshitaishvili, Marco Cova, Chris Kruegel, Giovanni Vigna
USENIX Security, 2013



+ classification



not a traditional query
not a traditional search result

Script summaries

how many “if” statements

how many “for” loops

...

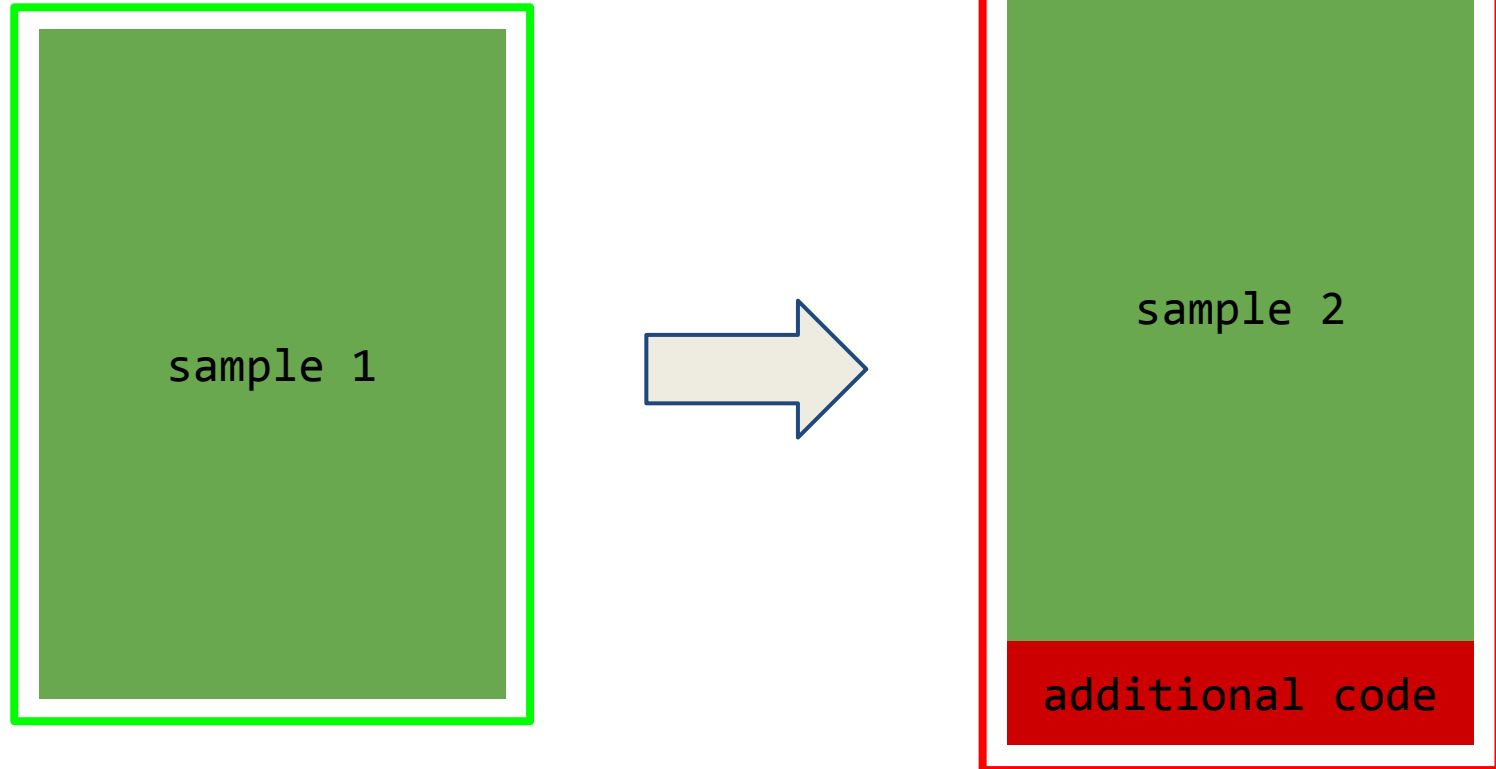
88-dimensional Euclidean space

k-nearest neighbor search

Classifying similar pairs

- Injection
 - Scripts that become malicious with additions

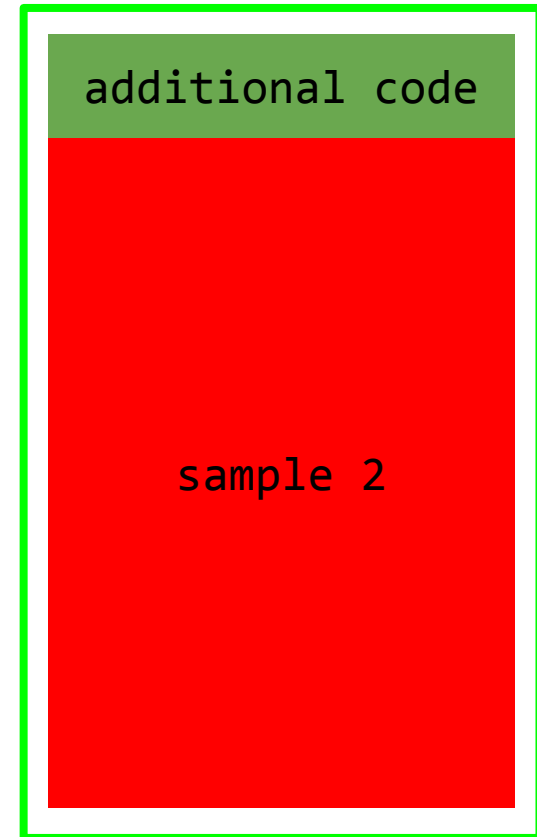
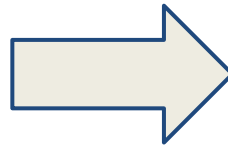
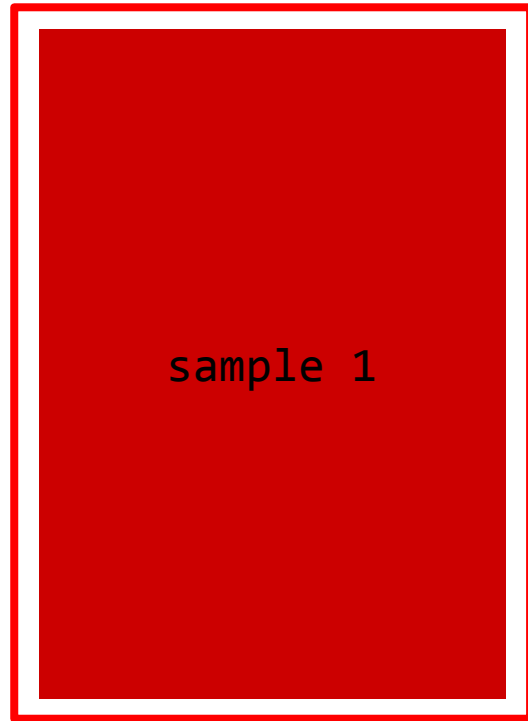
Injection



Classifying similar pairs

- Injection
 - Scripts that become malicious with additions
- Evasion
 - Scripts that become benign with control-flow changes

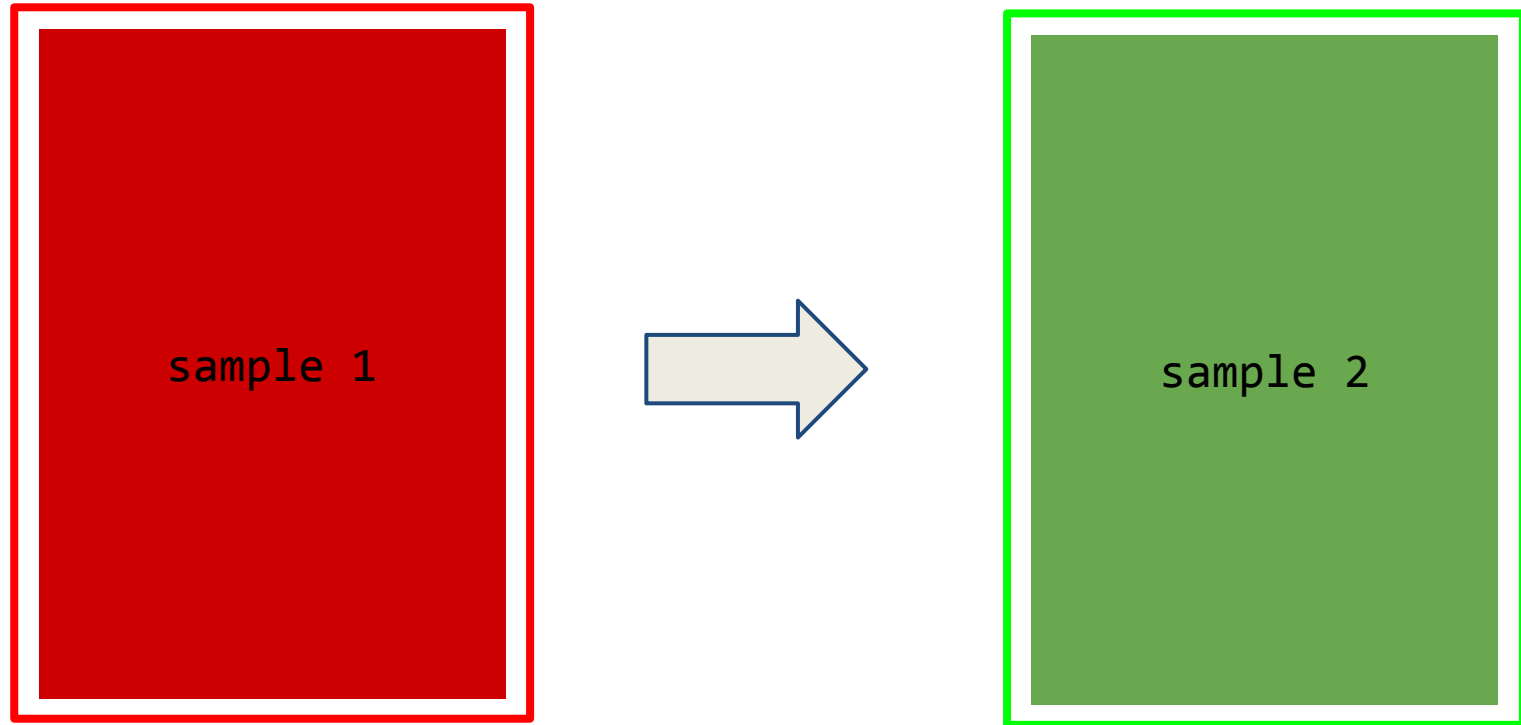
Evasion



Classifying similar pairs

- Injection
 - Scripts that become malicious with additions
- Evasion
 - Scripts that become benign with control-flow changes
- Data-dependency
 - Identical scripts with different classification

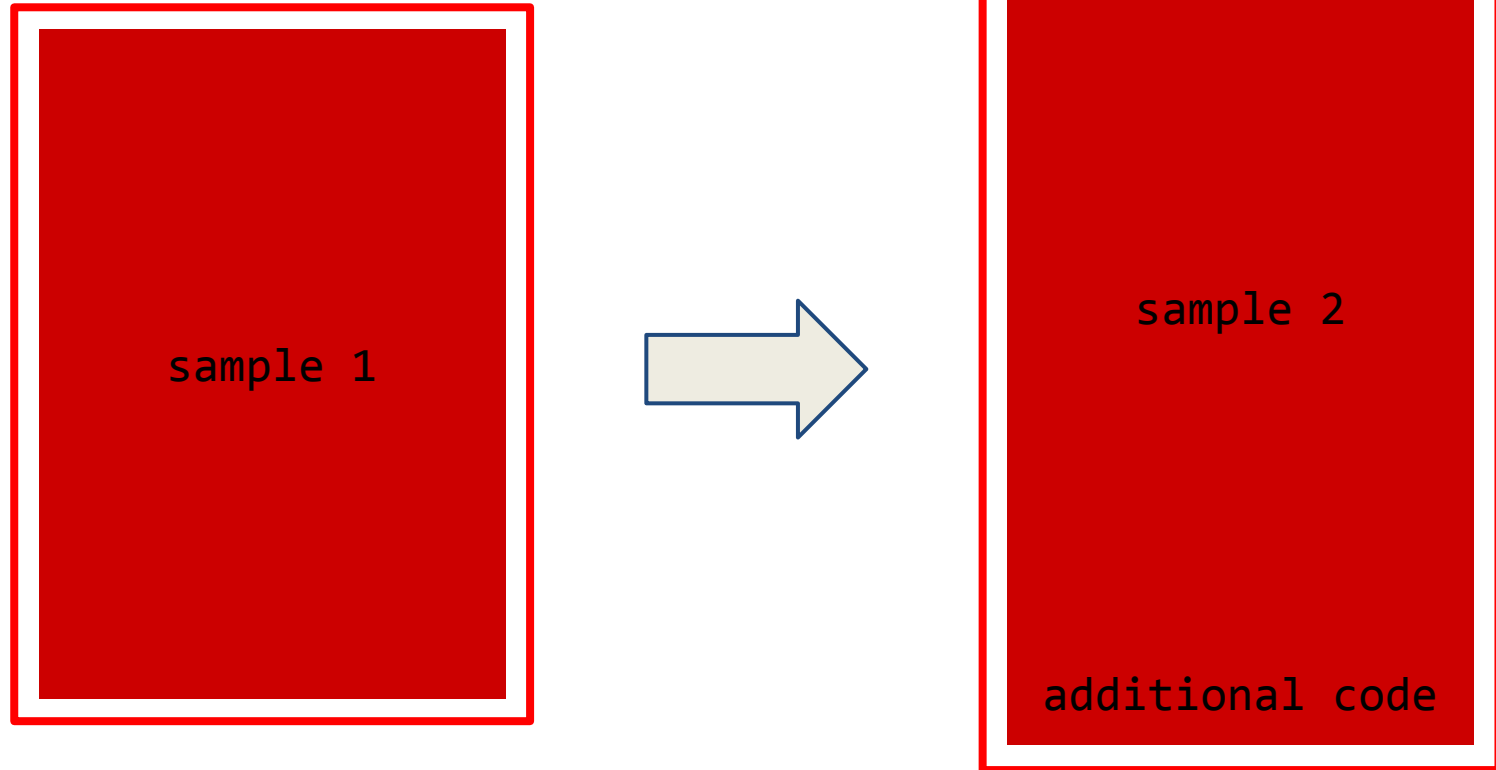
Data-dependency



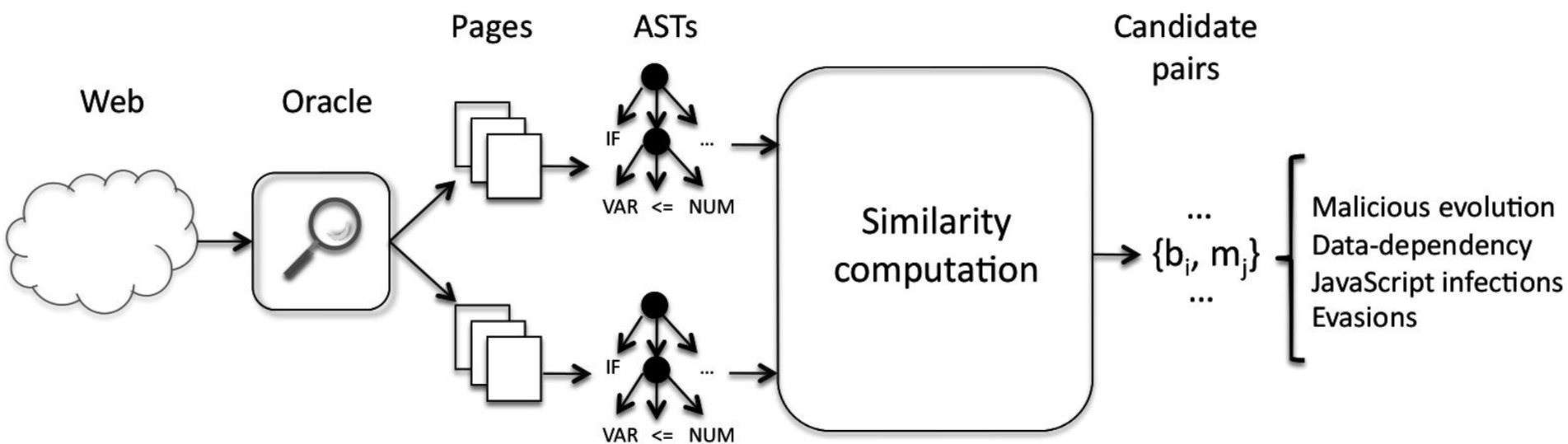
Classifying similar pairs

- Injection
 - Scripts that become malicious with additions
- Evasion
 - Scripts that become benign with control-flow changes
- Data-dependency
 - Identical scripts with different classification
- Evolution
 - Interesting to track for malicious-malicious pairs

Evolution



Architecture



Oracle

Revolver's input

- Any analysis system that can provide to Revolver:
 - JavaScript (even dynamically generated code)
 - Classification
- Wepawet in our experiments
 - Submit suspicious URLs at wepawet.cs.ucsb.edu
 - Every submission on Wepawet gets analyzed by Revolver in real-time

Abstract Syntax Tree (AST)

- Heavily obfuscated JavaScript
- All names are irrelevant
- Abstract the code as much as possible

Node sequences

- We break the structure of the tree and create sequences
- Nodes are integers representing node types

Sequence summary

- A statistical summary of node type occurrences

Similarities

- Deduplication
 - Identical scripts
- Approximate nearest neighbors
 - Based on sequence summary
 - Intuitively similar scripts have similar summaries
 - 88-dimensional Euclidean space and k-nearest neighbor search
- Directional similarities
 - Trying to match the malicious code

Experiments

- 6,468,623 web pages
 - 265,692 malicious pages
- 20,732,766 benign scripts
 - 705,472 unique benign ASTs
- 186,032 malicious scripts
 - 5,701 unique malicious ASTs

Results

Category	Similar Scripts	# Groups by malicious AST
JavaScript Injections	6,996	701
Data-dependencies	101,039	475
Evasions	4,147	155
General evolutions	2,490	273
Total	114,672	1,604

Evasions in the wild

JAVASCRIPT

```
// Malicious
function foo() {
  ...
  W6Kh6V5E4 = W6Kh6V5E4.replace(/\W/g, Bm2v5BSJE);
  ...
}

// Evasion
function foo(){
  ...
  var enryA = mxNEN+F7B07;
  F7B07 = eval;
  {}
  enryA = F7B07('enryA.rep' + 'lace(/\W/g,CxFHg));
  ...
}
```

JAVASCRIPT

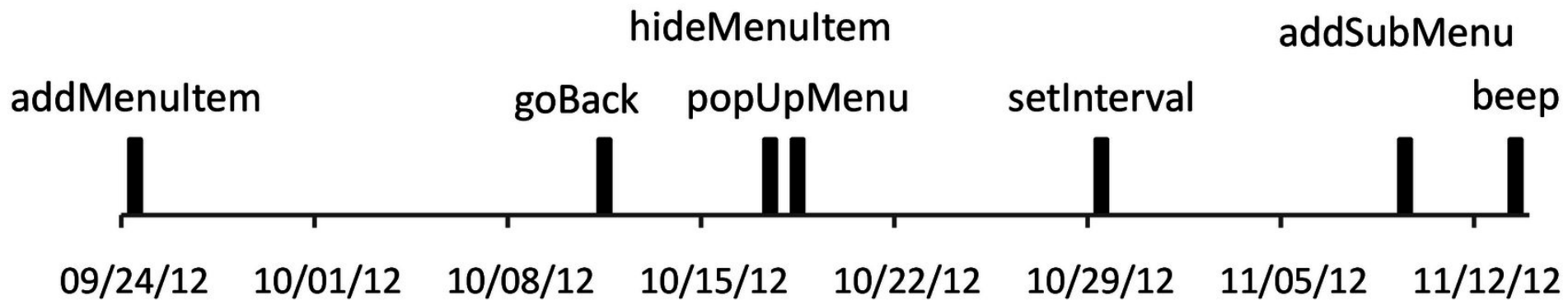
```
if((app.setInterval+/**/"" )["indexOf"](aa)!=-1){
a=/**/target.creationDate.split('|')[0];}
```

Evasions in the wild

JAVASCRIPT

// Malicious`0lhG='evil_code'``wTGB4=eval``wTGB4(0lhG)`*// Evasion*`0lhG='evil_code'``wTGB4="this"["eval"]``wTGB4(0lhG)`

Attackers' reactions



Limitations for Revolver

- No similarities
- Serve evasion before anything else
- Still need to analyze evasion and patch honeyclient manually

