MIS|TI™ PRESENTS

InfoSecWorld

Conference & Expo 2018

NO FILE REQUIRED: THE EMERGENCE OF FILELESS ATTACKS

Christopher Kruegel

CEO

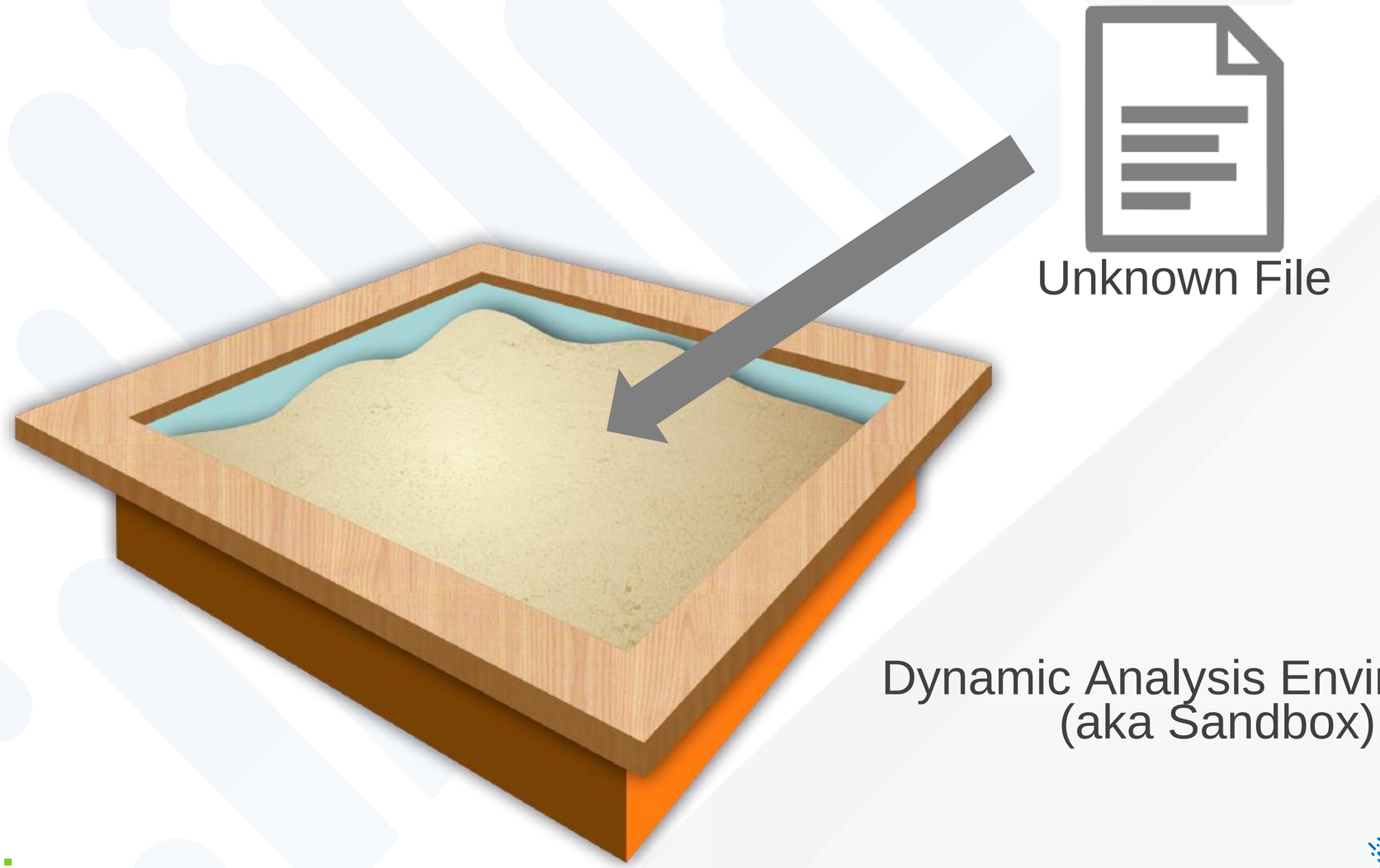
Lastline, Inc.

MOST POPULAR THREAT VECTORS

Email

Web Content

ZERO DAY THREAT DETECTION



Unknown File

Dynamic Analysis Environment
(aka Sandbox)

lastline

InfoSecWorld
Conference & Expo 2018

WHAT FILES ARE ANALYZED?

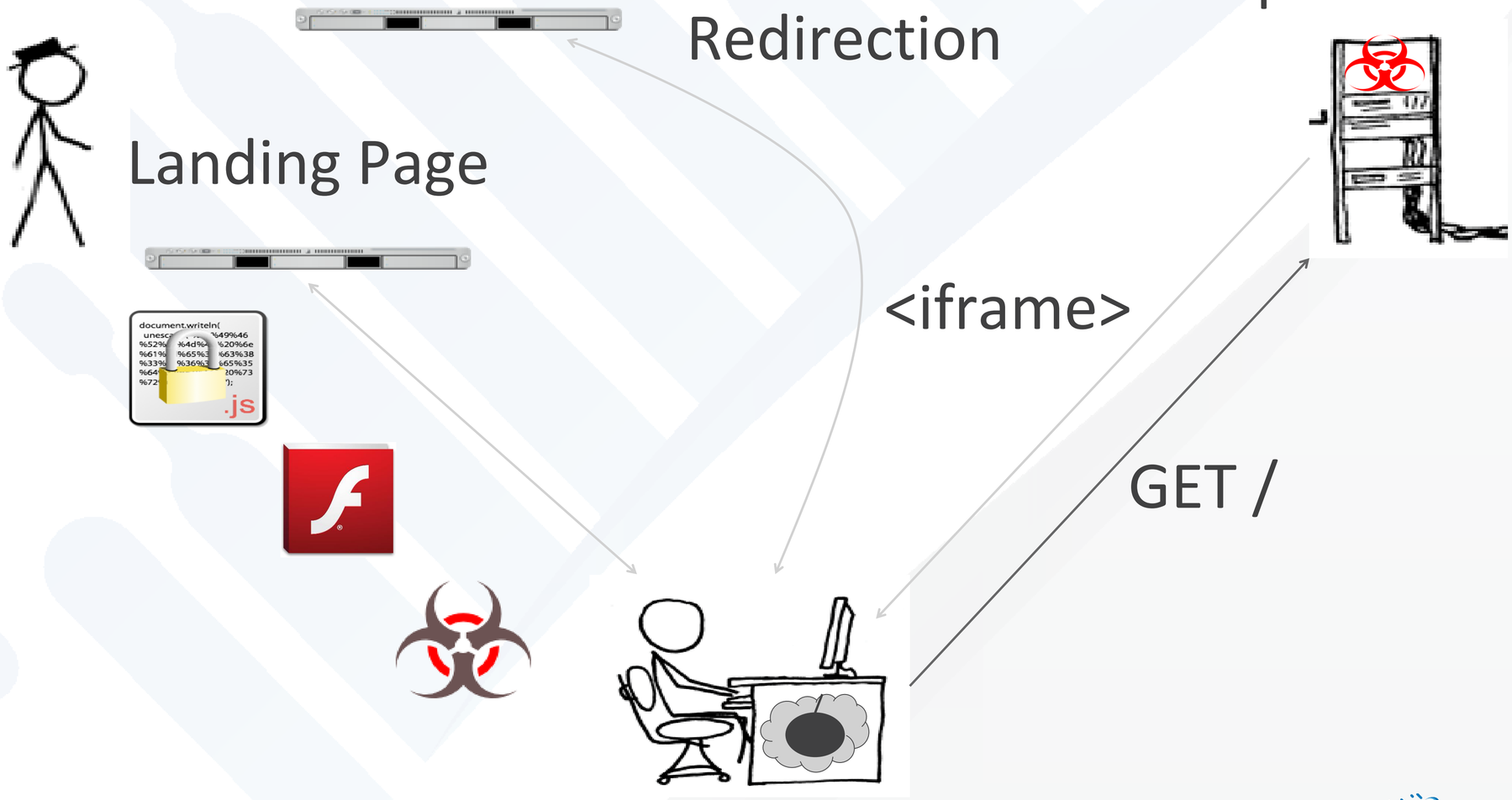
Attachments

- documents, archives, programs, …

… ?

Maybe the malware that is downloaded after the exploit?

DRIVE-BY-DOWNLOAD ATTACK



RIG EXPLOIT KIT

Filter: http.request or http.response | Expression... | Clear | Apply

No.	Source	Destination	Protocol	Info
4	victim	monteyplaya.com	HTTP	GET / HTTP/1.1
117	victim	nitrindo.lasestrellas	HTTP	GET /engine/classes/masha/masha.js HTTP/1.1
139	monteyplaya.com	victim	HTTP	HTTP/1.1 200 OK (text/html)
141	nitrindo.lasestrellas	victim	HTTP	HTTP/1.1 200 OK (text/javascript)
149	victim	lex.modernlily.info	HTTP	GET /?tuif=5184&biw=Mozilla.125uz102.406y7u4o3&b
153	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (text/html)
155	victim	lex.modernlily.info	HTTP	POST /?br_fl=2112&q=wX_QMvXcJwDQDYbGMvrESLtCNknQ
233	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (text/html)
236	victim	lex.modernlily.info	HTTP	GET /?yus=Microsoft_Edge.80ik95.406g9m5x4&q=wHnQ
264	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-shockwave-flash)
270	victim	lex.modernlily.info	HTTP	GET /?biw=SeaMonkey.124fs99.406k4n0p0&q=z3vQMvXc.
321	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-msdownload)
327	victim	spotsbill.com	HTTP	GET /find.php?g=2131375994&k=Fg5b79VFfJhHAZNmAbi
731	spotsbill.com	victim	HTTP	HTTP/1.1 200 OK (text/html)

Source: http://www.broadanalysis.com/2017/02/04/rig-exploit-kit-via-afraidgate-delivers-locky-ransomware-from-194-87-93-11/

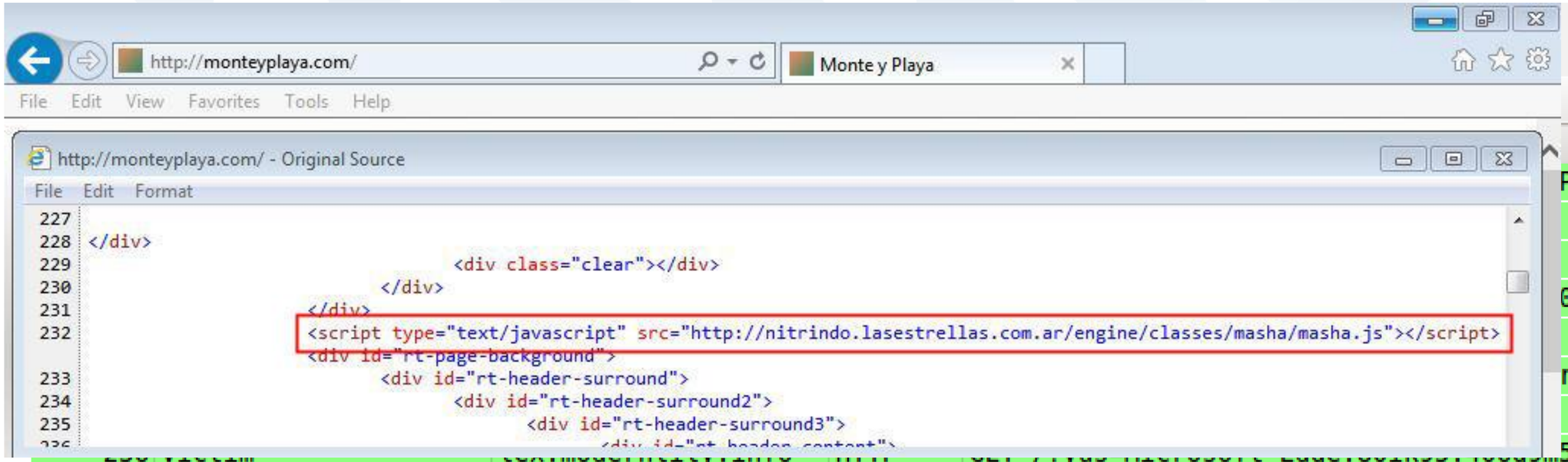
RIG EXPLOIT KIT

Filter: http.request or http.response | Expression... | Clear | Apply

No.	Source	Destination	Protocol	Info
4	victim	monteyplaya.com	HTTP	GET / HTTP/1.1
117	victim	nitrindo.lasestrellas	HTTP	GET /engine/classes/masha/masha.js HTTP/1.1
139	monteyplaya.com	victim	HTTP	HTTP/1.1 200 OK (text/html)
141	nitrindo.lasestrellas	victim	HTTP	HTTP/1.1 200 OK (text/javascript)
149	victim	lex.modernlily.info	HTTP	GET
153	lex.modernlily.info	victim	HTTP	HTT
155	victim	lex.modernlily.info	HTTP	POS
233	lex.modernlily.info	victim	HTTP	HTT
236	victim	lex.modernlily.info	HTTP	GET
264	lex.modernlily.info	victim	HTTP	HTT
270	victim	lex.modernlily.info	HTTP	GET
321	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-msdownload)
327	victim	spotsbill.com	HTTP	GET /find.php?g=2131375994&k=Fg5b79VFfJhHAZNmAbi
731	spotsbill.com	victim	HTTP	HTTP/1.1 200 OK (text/html)

Request to compromised page (`monteyplaya`) that includes iframe (`nitrindo`)

RIG EXPLOIT KIT



264	lex.modernlily.info	victim	HTTP	HTT
270	victim	lex.modernlily.info	HTTP	GET
321	lex.modernlily.info	victim	HTTP	HTT
327	victim	spotsbill.com	HTTP	GET
731	spotsbill.com	victim	HTTP	HTT

Request to compromised page (`monteyplaya`) that includes iframe (`nitrindo`)

RIG EXPLOIT KIT

Filter: http.request or http.response | Expression... | Clear | Apply

No.	Source	Destination	Protocol	Info
4	victim	monteyplaya.com	HTTP	GET / HTTP/1.1
117	victim	nitrindo.lasestrellas	HTTP	GET /engine/classes/masha/masha.js HTTP/1.1
139	monteyplaya.com	victim	HTTP	HTTP/1.1 200 OK (text/html)
141	nitrindo.lasestrellas	victim	HTTP	HTTP/1.1 200 OK (text/javascript)
149	victim	lex.modernlily.info	HTTP	GET /?tuif=5184&biw=Mozilla.125uz102.406y7u4o3&b
153	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (text/html)
155	victim	lex.modernlily.info	HTTP	POST /?br_fl=2112&q=wX_QMvXcJwDQDYbGMvrESLtCNknQ
233	lex.modernlily.info	victim	HTTP	HTT
236	victim	lex.modernlily.info	HTTP	GET
264	lex.modernlily.info	victim	HTTP	HTT
270	victim	lex.modernlily.info	HTTP	GET
321	lex.modernlily.info	victim	HTTP	HTT
327	victim	spotsbill.com	HTTP	GET
731	spotsbill.com	victim	HTTP	HTT

Redirection to landing page (`modernlily.info`)

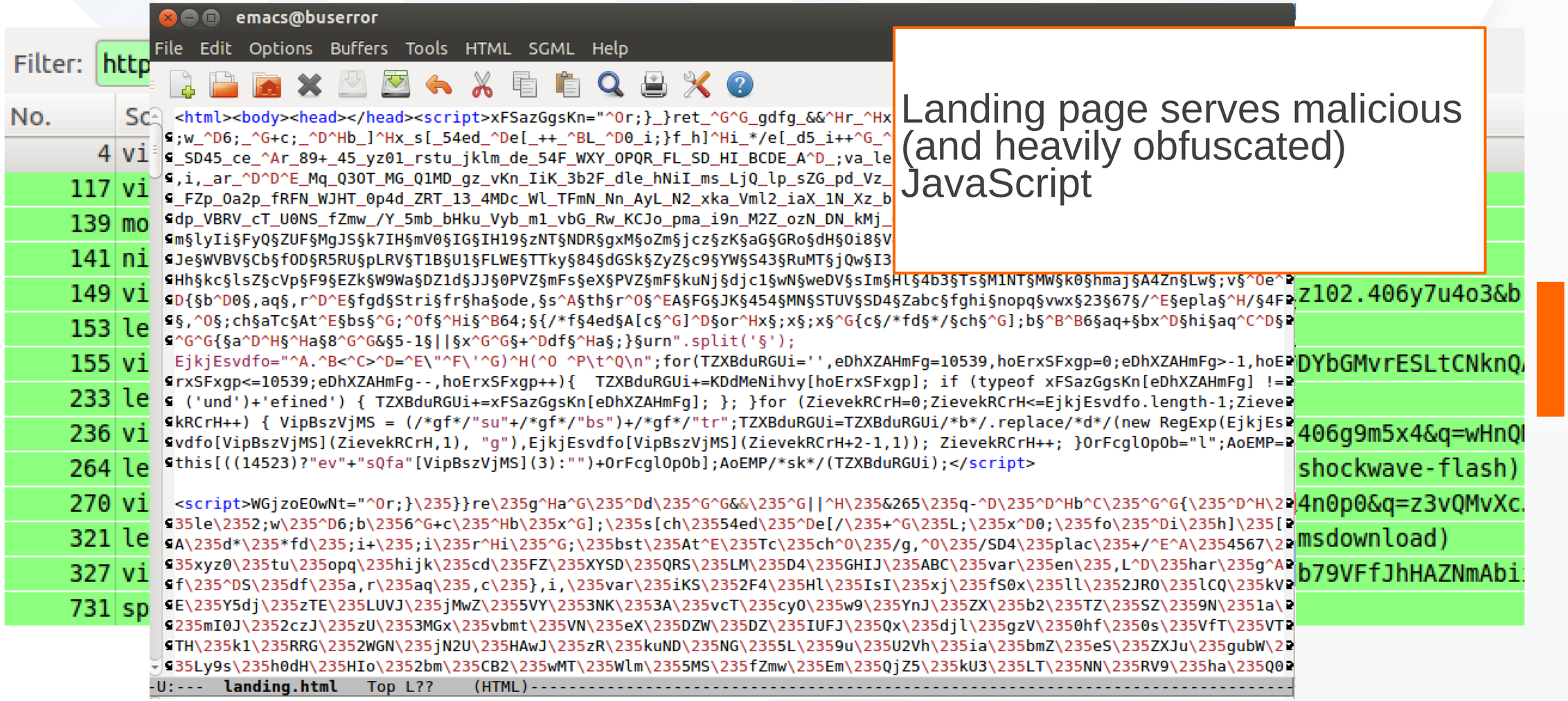
RIG EXPLOIT KIT

Filter: http.request or http.response | Express

No.	Source	Destination	Protocol	Info
4	victim	monteyplaya.com	HTTP	GET
117	victim	nitrindo.lasestrellas	HTTP	GET
139	monteyplaya.com	victim	HTTP	HTT
141	nitrindo.lasestrellas	victim	HTTP	HTT
149	victim	lex.modernlily.info	HTTP	GET /?tuif=5184&biw=Mozilla.125uz102.406y7u4o3&b
153	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (text/html)
155	victim	lex.modernlily.info	HTTP	POST /?br_fl=2112&q=wX_QMvXcJwDQDYbGMvrESLtCNknQ
233	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (text/html)
236	victim	lex.modernlily.info	HTTP	GET /?yus=Microsoft_Edge.80ik95.406g9m5x4&q=wHnQ
264	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-shockwave-flash)
270	victim	lex.modernlily.info	HTTP	GET /?biw=SeaMonkey.124fs99.406k4n0p0&q=z3vQMvXc.
321	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-msdownload)
327	victim	spotsbill.com	HTTP	GET /find.php?g=2131375994&k=Fg5b79VFfJhHAZNmAbi
731	spotsbill.com	victim	HTTP	HTTP/1.1 200 OK (text/html)

Landing page serves malicious (and heavily obfuscated) JavaScript

RIG EXPLOIT KIT



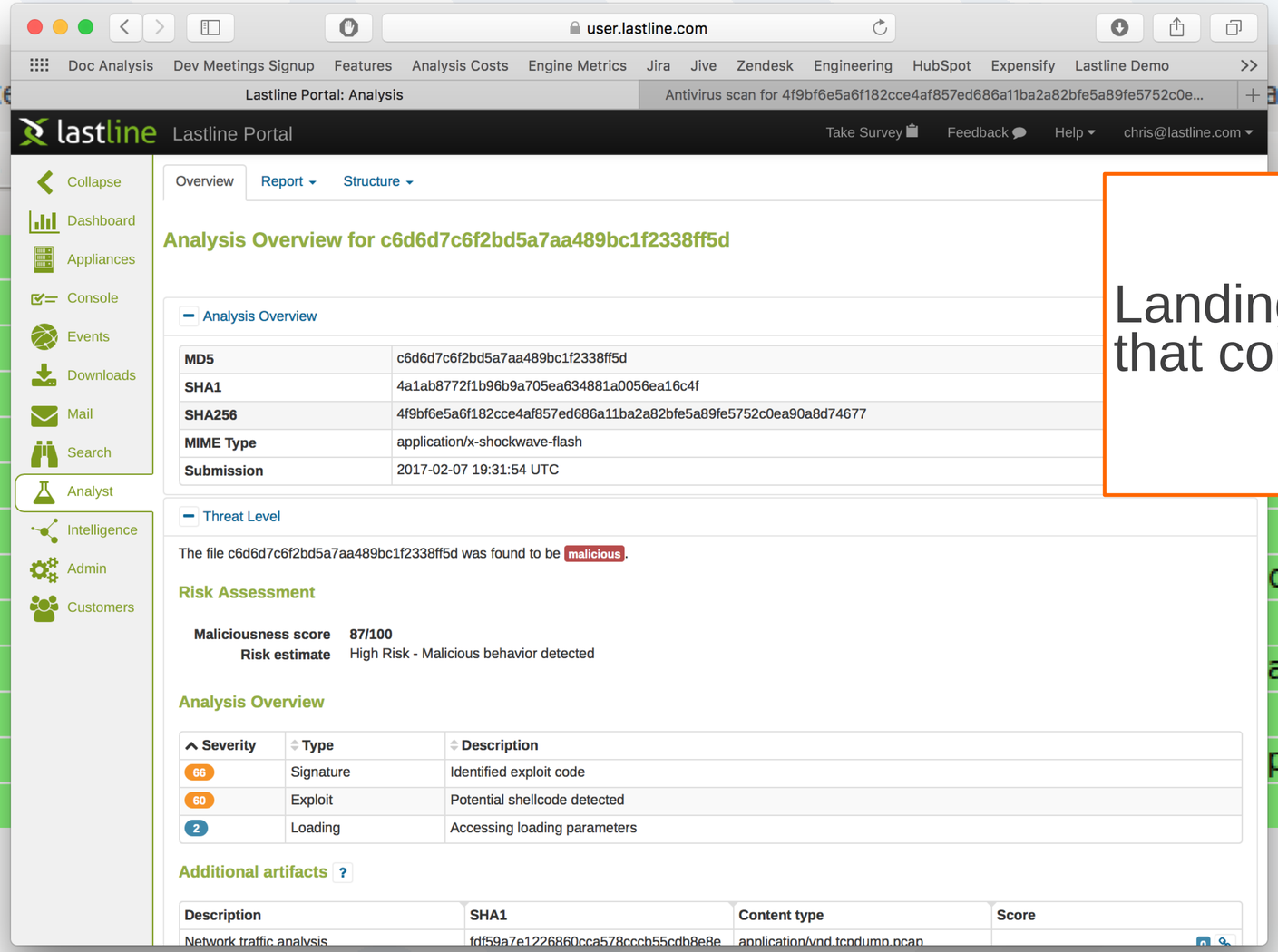
RIG EXPLOIT KIT

Filter: http.request or http.response | Expression... | Clear | Apply

No.	Source	Destination	Protocol	Info
4	victim	monteyplaya.com	HTTP	GET
117	victim	nitrindo.lasestrellas	HTTP	GET
139	monteyplaya.com	victim	HTTP	HTT
141	nitrindo.lasestrellas	victim	HTTP	HTT
149	victim	lex.modernlily.info	HTTP	GET
153	lex.modernlily.info	victim	HTTP	HTT
155	victim	lex.modernlily.info	HTTP	POS
233	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (text/html)
236	victim	lex.modernlily.info	HTTP	GET /?yus=Microsoft_Edge.80ik95.406g9m5x4&q=wHnQ
264	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-shockwave-flash)
270	victim	lex.modernlily.info	HTTP	GET /?biw=SeaMonkey.124fs99.406k4n0p0&q=z3vQMvXc.
321	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-msdownload)
327	victim	spotsbill.com	HTTP	GET /find.php?g=2131375994&k=Fg5b79VFfJhHAZNmAbi
731	spotsbill.com	victim	HTTP	HTTP/1.1 200 OK (text/html)

Landing page serves Flash file that contains exploits

RIG EXPLOIT KIT

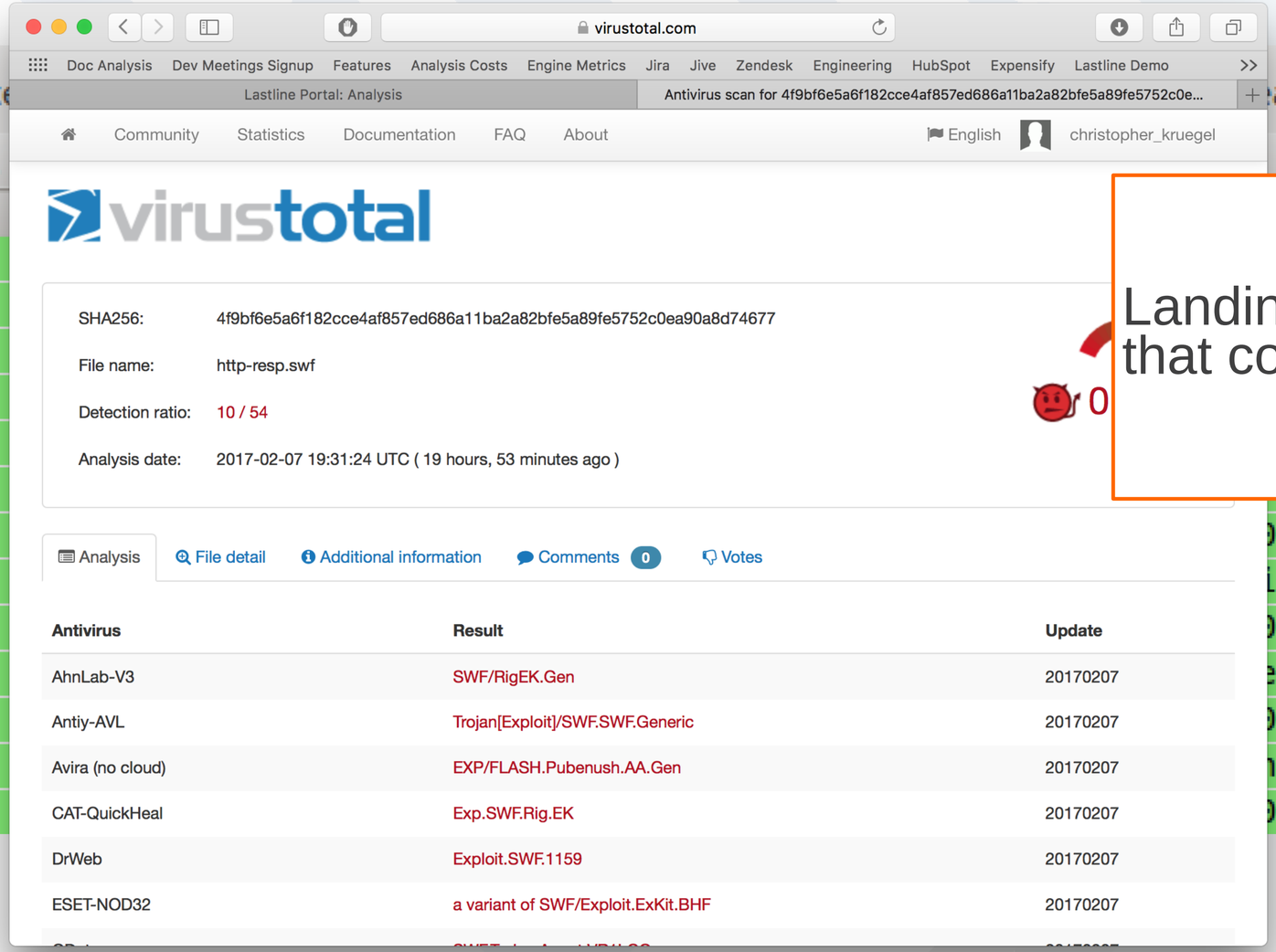


Landing page serves Flash file that contains exploits

lastline

InfoSecWorld Conference & Expo 2018

RIG EXPLOIT KIT



Landing page serves Flash file that contains exploits

lastline

InfoSecWorld Conference & Expo 2018

RIG EXPLOIT KIT

Filter: http.request or http.response | Expression... | Clear | Apply

No.	Source	Destination	Protocol	Info
4	victim	monteyplaya.com	HTTP	GET / HTTP/1.1
117	victim	nitrindo.lasestrellas	HTTP	GET /engine/classes/masha/masha.js HTTP/1.1
139	monteyplaya.com	victim	HTTP	HTT
141	nitrindo.lasestrellas	victim	HTTP	HTT
149	victim	lex.modernlily.info	HTTP	GET
153	lex.modernlily.info	victim	HTTP	HTT
155	victim	lex.modernlily.info	HTTP	POS
233	lex.modernlily.info	victim	HTTP	HTT
236	victim	lex.modernlily.info	HTTP	GET
264	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-shockwave-flash)
270	victim	lex.modernlily.info	HTTP	GET /?biw=SeaMonkey.124fs99.406k4n0p0&q=z3vQMvXc.
321	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-msdownload)
327	victim	spotsbill.com	HTTP	GET /find.php?g=2131375994&k=Fg5b79VFfJhHAZNmAbi
731	spotsbill.com	victim	HTTP	HTTP/1.1 200 OK (text/html)

After successful exploit, shellcode requests (obfuscated) malware

RIG EXPLOIT KIT

Filter: http.request or http.response | Expression... | Clear | Apply

No.	Source	Destination	Protocol	Info
4	victim	monteyplaya.com	HTTP	GET / HTTP/1.1
117	victim	nitrindo.lasestrellas	HTTP	GET /engine/classes/masha/masha.js HTTP/1.1
139	monteyplaya.com	victim	HTTP	HTTP/1.1 200 OK (text/html)
141	nitrindo.lasestrellas	victim	HTTP	HTT
149	victim	lex.modernlily.info	HTTP	GET
153	lex.modernlily.info	victim	HTTP	HTT
155	victim	lex.modernlily.info	HTTP	POS
233	lex.modernlily.info	victim	HTTP	HTT
236	victim	lex.modernlily.info	HTTP	GET
264	lex.modernlily.info	victim	HTTP	HTT
270	victim	lex.modernlily.info	HTTP	GET
321	lex.modernlily.info	victim	HTTP	HTTP/1.1 200 OK (application/x-msdownload)
327	victim	spotsbill.com	HTTP	GET /find.php?g=2131375994&k=Fg5b79VFfJhHAZNmAbi
731	spotsbill.com	victim	HTTP	HTTP/1.1 200 OK (text/html)

Malware initiates command and control connection (`spotsbill.com`)

FILE-LESS WEB THREATS

Malware program was already obfuscated

- but at least we had the Flash file

What if there is no Flash file on the wire?

- embed Flash file in obfuscated JavaScript
- browser vulnerabilities triggered directly from JavaScript

What if the shellcode does not load additional malware?

- directly manipulate browser process, never drop anything

FILE-LESS WEB THREATS

Reputation information

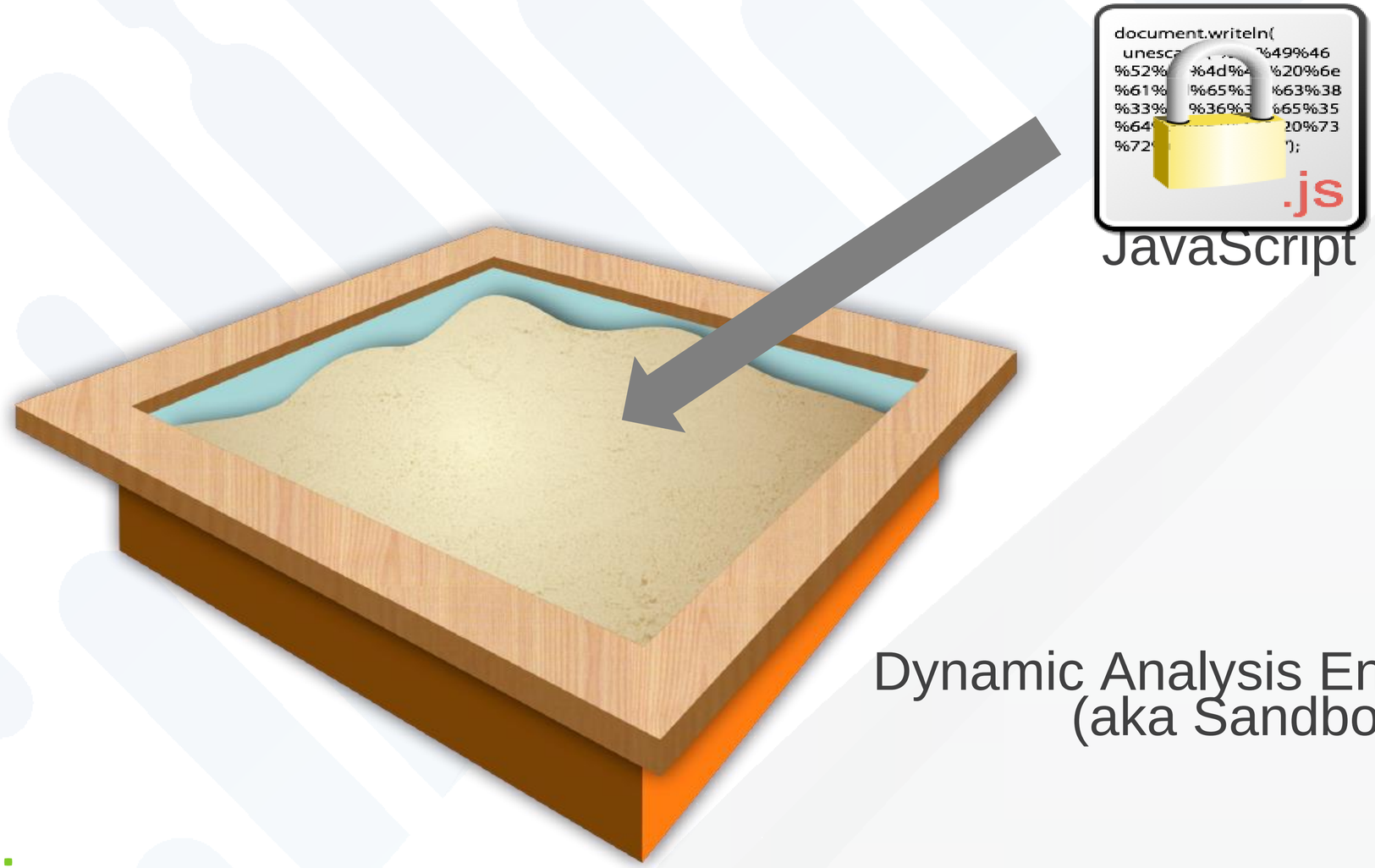
- blacklists, threat intelligence, …

Signatures

- exploit signatures, URL patterns, ...

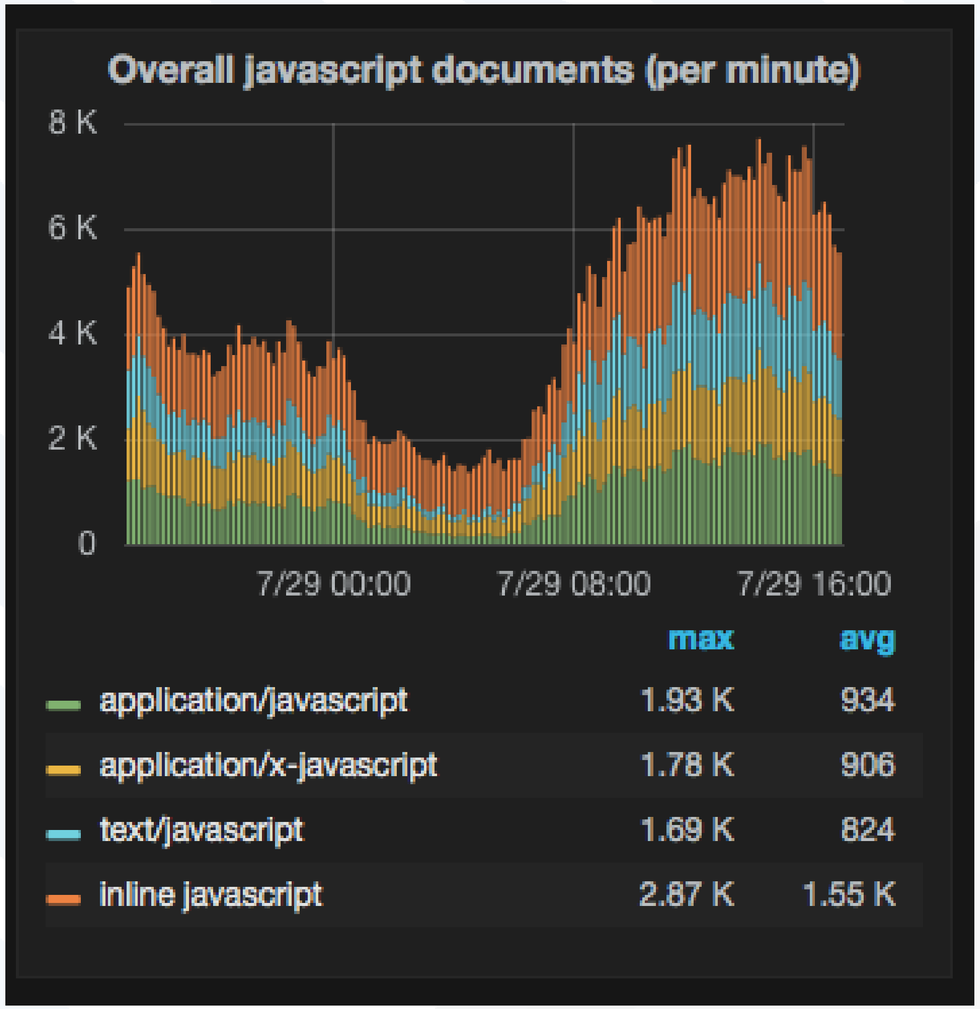
➔ Cannot handle zero day threats

SANDBOX FOR JAVASCRIPT?

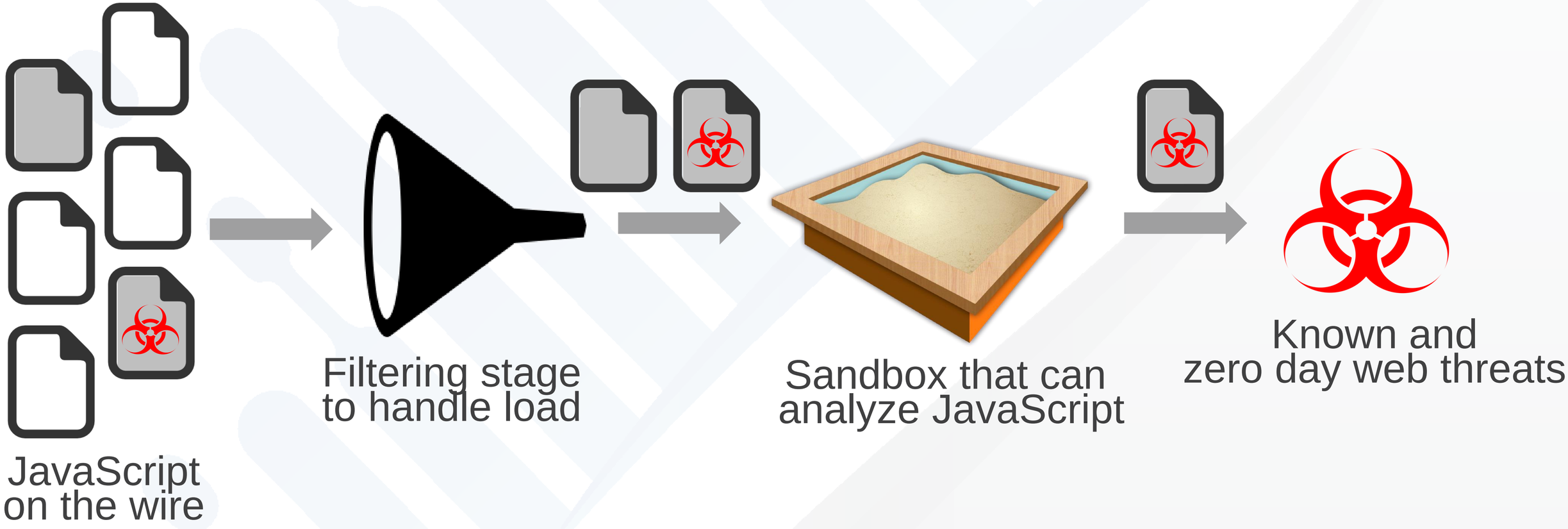


Dynamic Analysis Environment
(aka Sandbox)

LOTS OF JAVASCRIPT ON THE WIRE!



WEB THREAT DETECTION PIPELINE



lastline

InfoSecWorld
Conference & Expo 2018

WEB THREAT DETECTION PIPELINE

Funnel (filter layers using anomaly detection)

- anomalies due to code injection
  - hidden iframes, out-of-place tags, …
- anomalies due to obfuscation
  - encodings, character distribution, specific JS functions, …
- exploit indicators
  - structural similarities, signatures, …

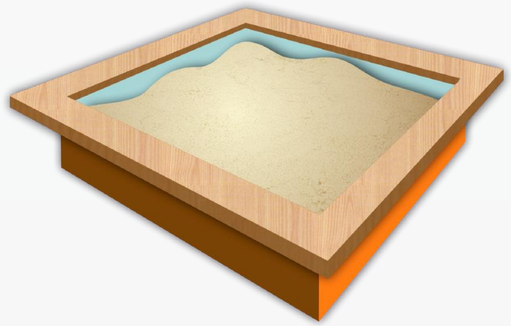
lastline

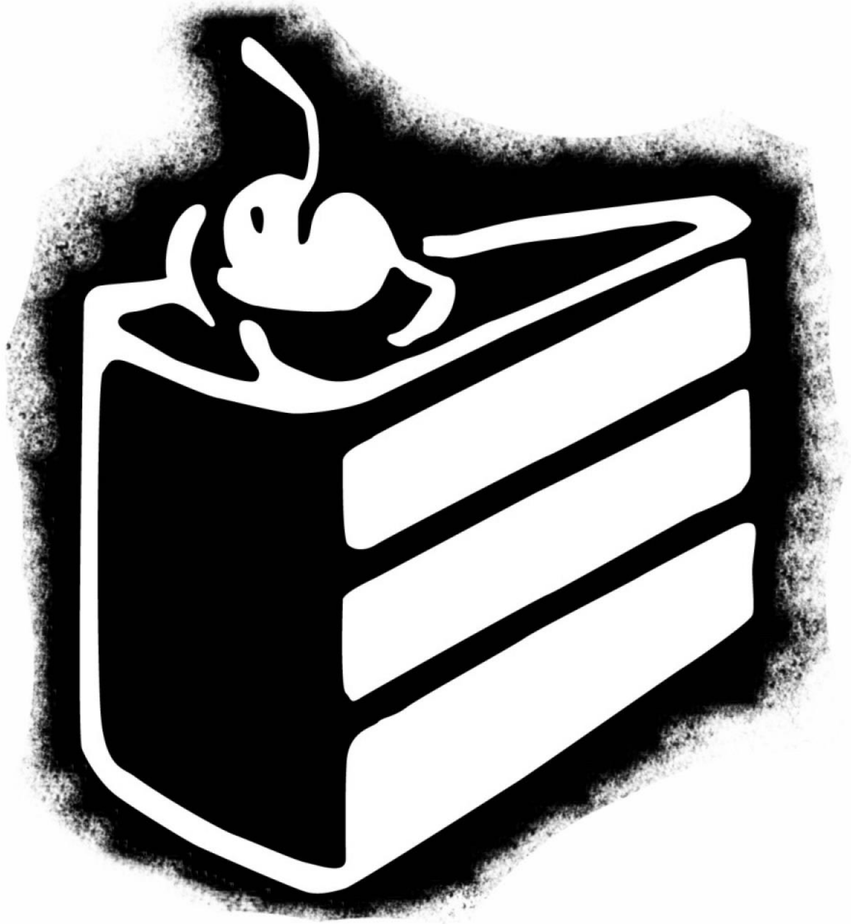
InfoSecWorld Conference & Expo 2018

WEB THREAT DETECTION PIPELINE

JavaScript sandbox (instrumented browser)

- process behavior (drop and execute)
- heap spray
- check for shellcode in strings
- environment checks
- suspicious function arguments



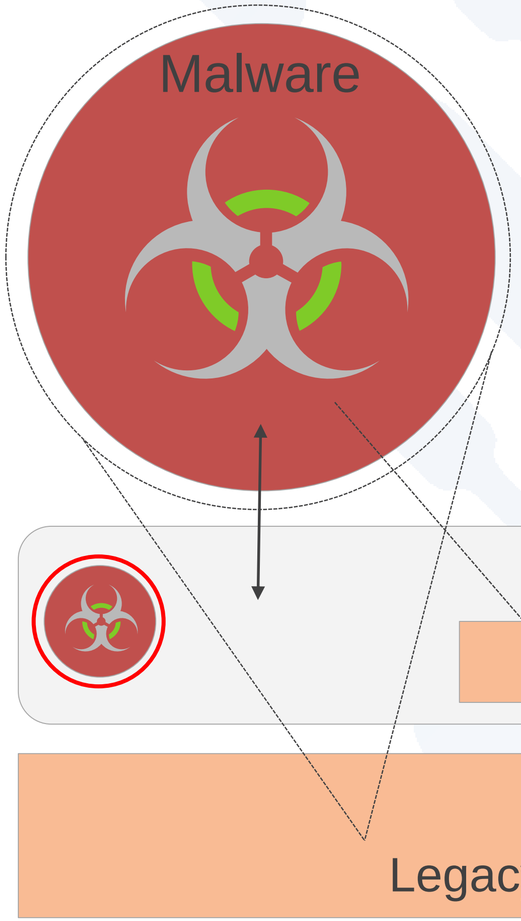


the ~~cake~~ is a lie!

One Simple Trick

FUSE – FULL SYSTEM EMULATION

FUSE – DEEP VISIBILITY INTO MALWARE



Visibility limited to interactions with the OS

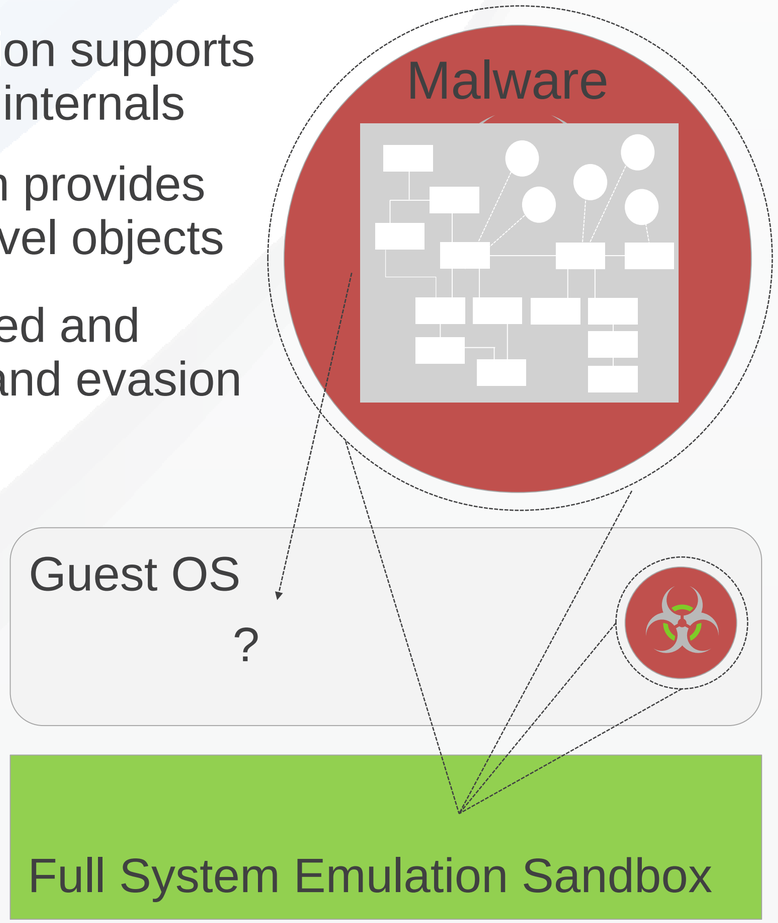
Visibility limited to user-space objects

Virtualization artifacts in Guest OS allow malware to fingerprint and evade detection

Deep content Inspection supports visibility into malware internals

Full-system emulation provides visibility into kernel-level objects

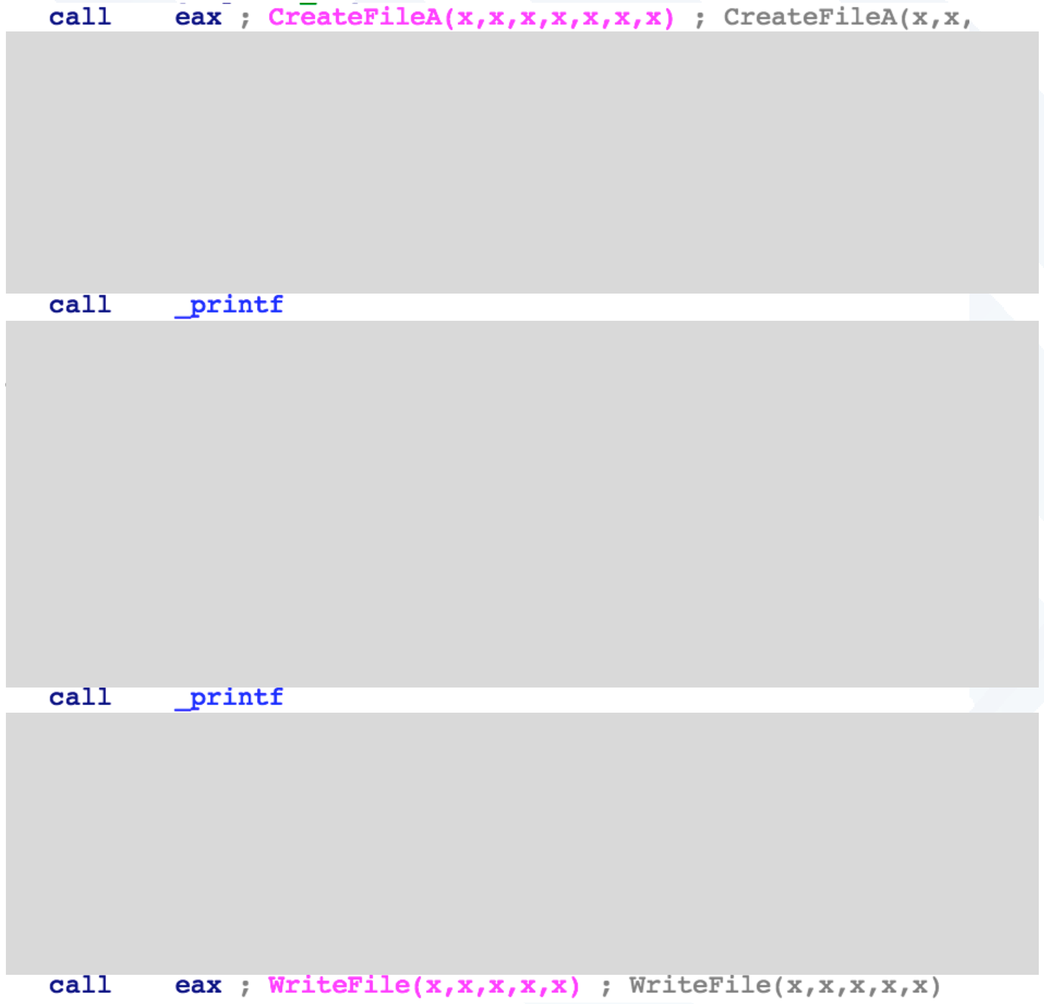
Guest OS is unmodified and resists fingerprinting and evasion



lastline

InfoSecWorld Conference & Expo 2018

FUSE – DEEP VISIBILITY INTO MALWARE



```
call    eax ; CreateFileA(x,x,x,x,x,x,x) ; CreateFileA(x,x,
call    _printf
call    _printf
call    eax ; WriteFile(x,x,x,x,x) ; WriteFile(x,x,x,x,x)
```

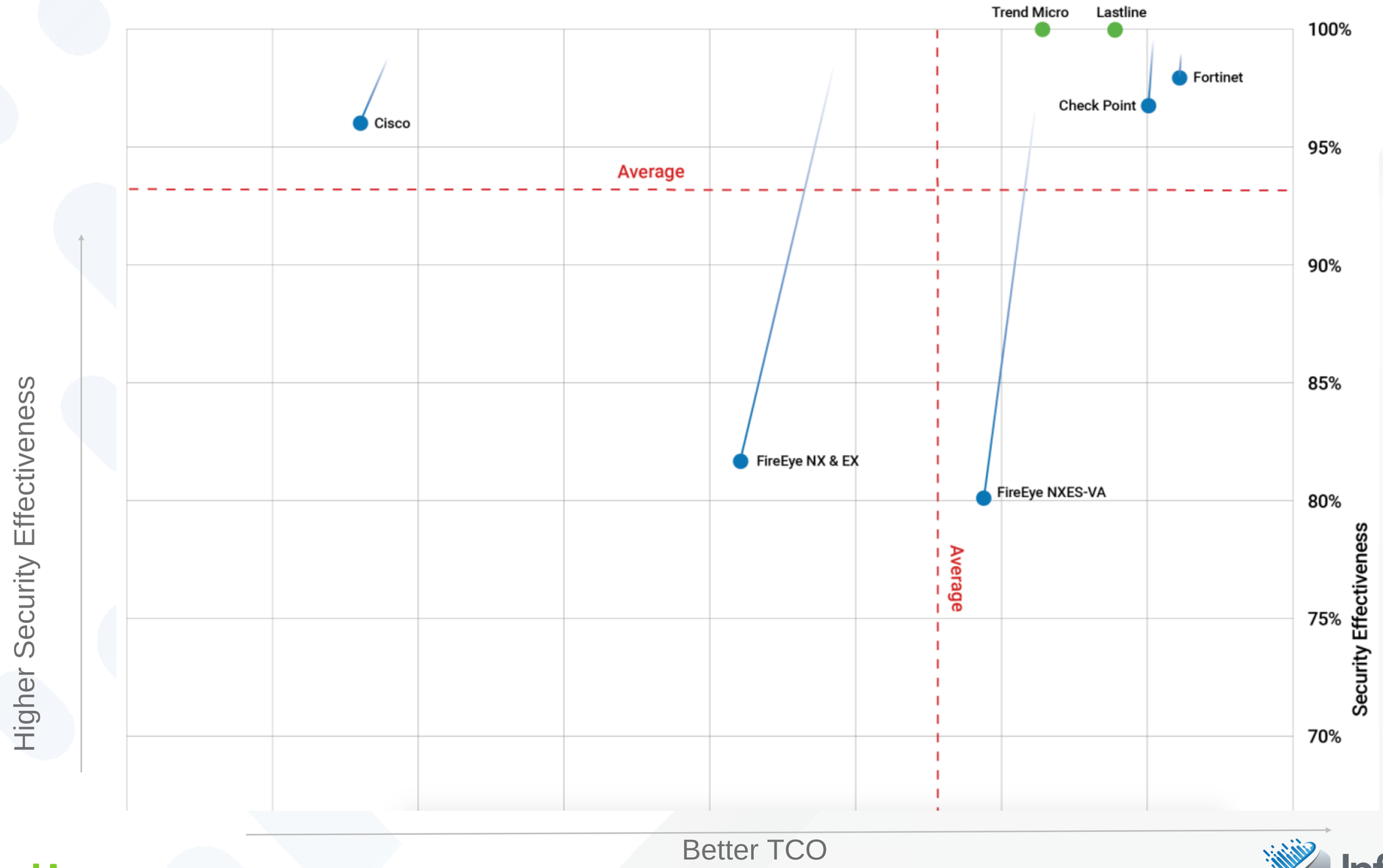
```
call    eax ; CreateFileA(x,x,x,x,x,x,x) ; CreateFileA(x,x,
sub     esp, 1Ch
mov     [ebp+hFile], eax
cmp     [ebp+hFile], 0FFFFFFFFh
jnz     short loc_4016A4
mov     eax, [ebp+var_90]
mov     eax, [eax+4]
add     eax, 4
mov     eax, [eax]
mov     [esp+4], eax
mov     dword ptr [esp], offset aTerminalFailur ; "Terminal
call    _printf
mov     eax, 1
jmp     loc_401785
-------------------------------------------------------------

                        ; CODE XREF: _main+11A↑j
mov     eax, [ebp+var_90]
mov     eax, [eax+4]
add     eax, 4
mov     eax, [eax]
mov     [esp+8], eax
mov     eax, [ebp+nNumberOfBytesToWrite]
mov     [esp+4], eax
mov     dword ptr [esp], offset aWritingDBytesT ; "Writing
mov     [ebp+var_88], 1
call    _printf
mov     dword ptr [esp+10h], 0 ; lpOverlapped
lea     eax, [ebp+NumberOfBytesWritten]
mov     [esp+0Ch], eax  ; lpNumberOfBytesWritten
mov     eax, [ebp+nNumberOfBytesToWrite]
mov     [esp+8], eax    ; nNumberOfBytesToWrite
lea     eax, [ebp+Buffer]
mov     [esp+4], eax    ; lpBuffer
mov     eax, [ebp+hFile]
mov     [esp], eax      ; hFile
mov     eax, ds:__imp__WriteFile@20 ; WriteFile(x,x,x,x,x)
call    eax ; WriteFile(x,x,x,x,x) ; WriteFile(x,x,x,x,x)
```

lastline

InfoSecWorld Conference & Expo 2018

NSS BDS GROUP TEST EVALUATION



lastline

InfoSecWorld Conference & Expo 2018

DEALING WITH EVASIVE, FILE-LESS THREATS

EVASION: SCOPE HANDLING

```
function foo() {
     ... // W6Kh6V5E4 is filled with non-alpha data
     Bm2v5BSJE="";
     W6Kh6V5E4 = W6Kh6V5E4.replace(/\W/g,Bm2v5BSJE);
     ... // W6Kh6V5E4 now contains valid JavaScript
}
```

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

EVASION: INTERPRETER IDIOMS

```
OlhG='evil_code'
wTGB4=eval
wTGB4(OlhG)
```

```
OlhG='evil_code'
wTGB4="this"["eval"] // Only works in Adobe's JS
wTGB4(OlhG)
```

EVASION: EXCEPTION PATHS

```
function deobfuscate(){
    // Define variable xorkey
    // and compute its value
    for(...) {
      ... // decrypt with xorkey
    }
    eval(deobfuscated_string);
}

try {
  eval(deobfuscate();)
}
catch (e){
  alert('err');
}
```

```
function deobfuscate(){
  try { ... // is xorkey defined? }
  catch(e){ xorkey=0; }
  ... // Compute value of xorkey
  VhplKO8 += 1; // throws exception
  for(...) {
    ... // decrypt with xorkey
  }
  eval(deobfuscated_string);
}

try { eval(deobfuscate();) } // 1st
catch (e){
  // Variable VhplKO8 is not defined
  try {
    VhplKO8 = 0; // define variable
    eval(deobfuscate();); // 2nd
  } catch (e){ alert('err'); }
}
```

EVASION: LIBERAL CONFIGURATION

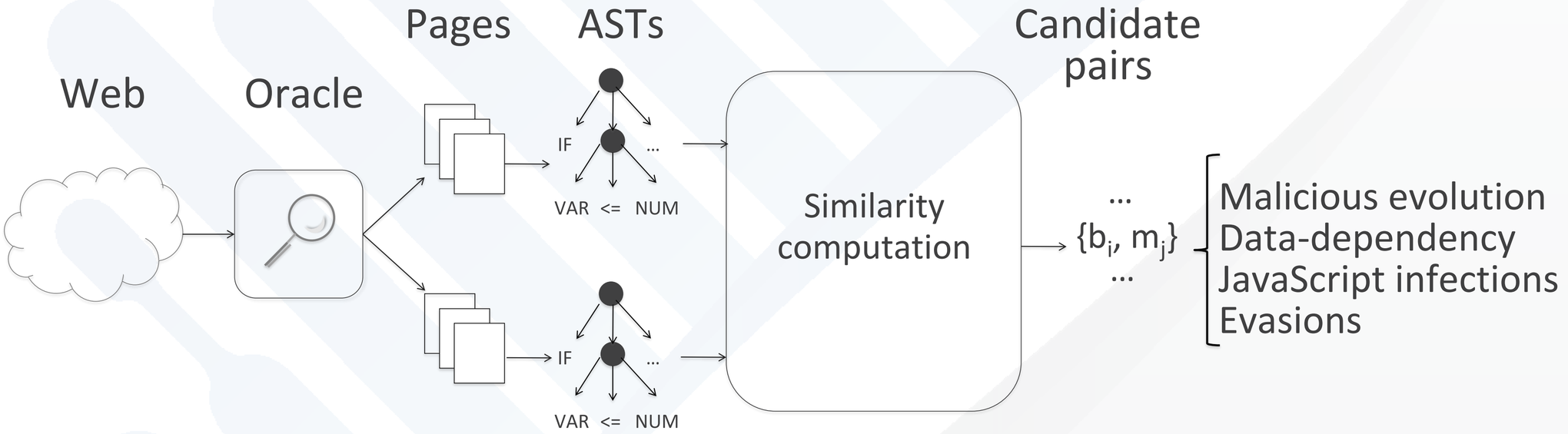
```
var nop= "%uyt9yt2yt9yt2";
var nop= (nop.replace(/yt/g,""));
var sc0= "%ud5db%uc9c9%u87cd...";
var sc1= "%"+"yutianu"+
    "ByutianD"+ ...;
var sc1=
    (sc1.replace(/yutian/g,""));
var sc2= "%"+"u"+"54"+"FF”+"%u”
    +"BE"+...+"A"+"8"+"E"+"E";
var sc2=
    (sc2.replace(/yutian/g,""));
var sc= unescape(nop+sc0+sc1+sc2);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

```
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”+"%u”
      +"BE"+...+"A"+"8"+"E"+"E";
  var sc2=
      (sc2.replace(/yutian/g,""));
  var sc= unescape(nop+sc0+sc1+sc2);
}
```

DETECTING EVASION: CHALLENGES

- Code is obfuscated
- Code is generated on-the-fly
- Code might probe for arcane versions of a browser
- Not all code changes are relevant

REVOLVER


Web
Oracle
Pages
ASTs
IF
...
VAR <= NUM
IF
...
VAR <= NUM
Similarity computation
Candidate pairs
...
{b_i, m_j}
...
Malicious evolution
Data-dependency
JavaScript infections
Evasions

APPROACH

1. Extract ASTs of executed scripts
2. An oracle determines which ASTs are malicious
3. Compare every pair of ASTs where at least one ASTs is malicious, and determine if the similarity is above a certain threshold
4. If similarity exists between two malicious scripts, difference is flagged as an evolution
5. If similarity exists between a malicious script and a benign one, difference is flagged as one of
    1. Data dependency
    2. Injection
    3. Evasion

CLASSIFICATION

Data-dependency: categorizes script differences that are associated with transforming data into code

- same packers usually produce different code: if generating code is same and generated code is very different, do not flag as evasion

Injection: categorizes script differences that are due to addition of code to a previously-benign script

- site gets compromised and attacker adds code to well-known JavaScript libraries (e.g., jQuery)

Evasion: categorizes script differences that are mostly composed of control-flow nodes added to the previously-malicious script

- control-flow decisions are made to avoid executing the malicious functionality

EVALUATION: EVASION

- Collected 6,468,623 pages (of which 265,692 malicious)
- Extracted 20,732,766 benign and 186,032 malicious scripts
- Derived 705,472 unique ASTs and 5,701 malicious ASTs
- For each benign AST, found ~70 malicious neighbors
- Computed 208K candidate pairs
  - 6,996 Injections (701 classes)
  - 101,039 Data dependencies (475 classes)
  - 4,147 Evasions (155 classes)
  - 2,490 Evolutions (273 classes)

MIS|TI™ PRESENTS

InfoSec World

Conference & Expo 2018

THANK YOU

PLEASE FILL OUT YOUR EVALUATIONS!

Christopher Kruegel

CEO

Lastline, Inc.