Winning the Online Banking War

Sean Park

Senior Malware Scientist

TrendMicro

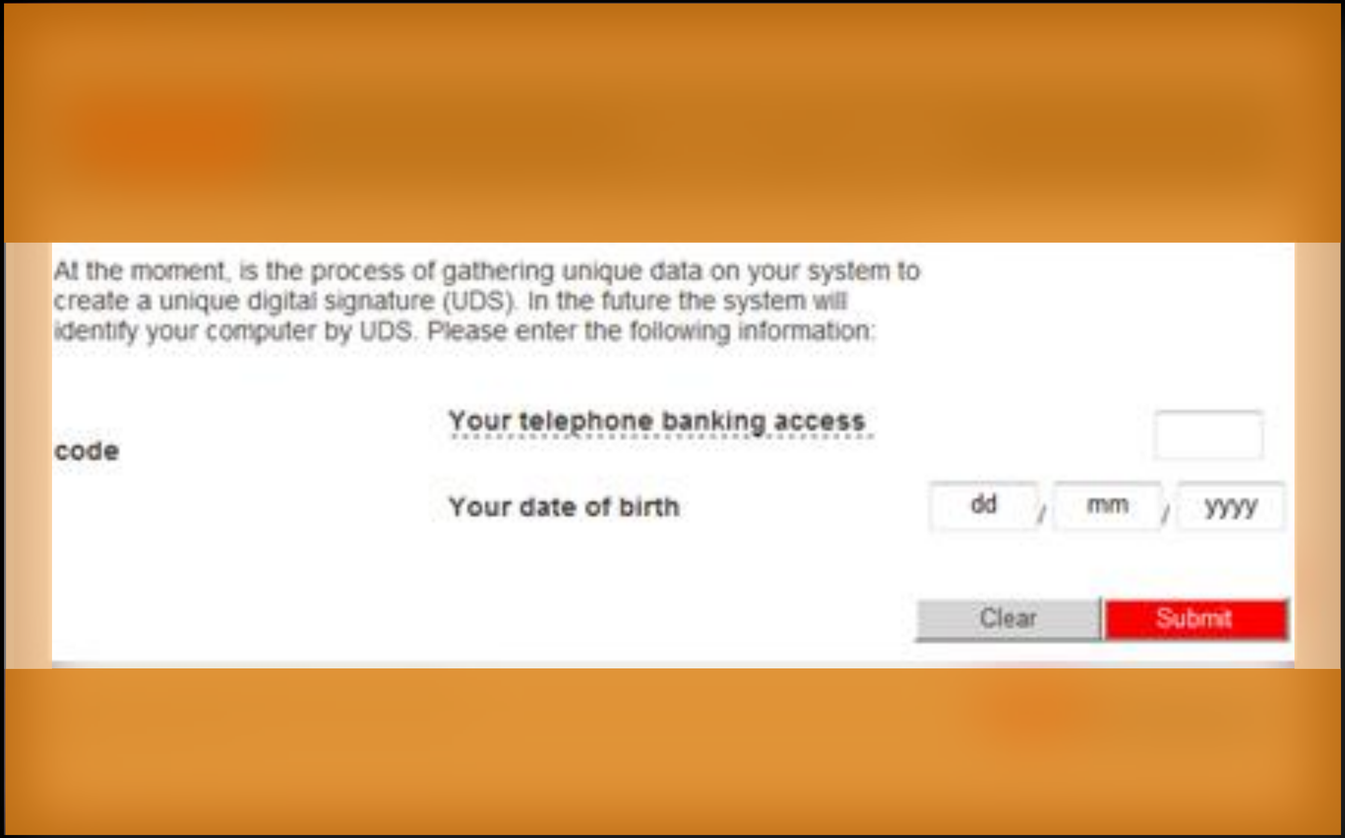
spark@trendmicro.com

Overview

- Examples of Web Injects
- Basic Concept
- Attack and Defenses
  - DOM Stealth
  - Replay Attack
  - MIPS Forgery
  - DOM Rootkit
  - MIPS Blocking
  - ZKP
  - Live Demo for Each Attack

Web Injects

Sign In – Phone Banking Code

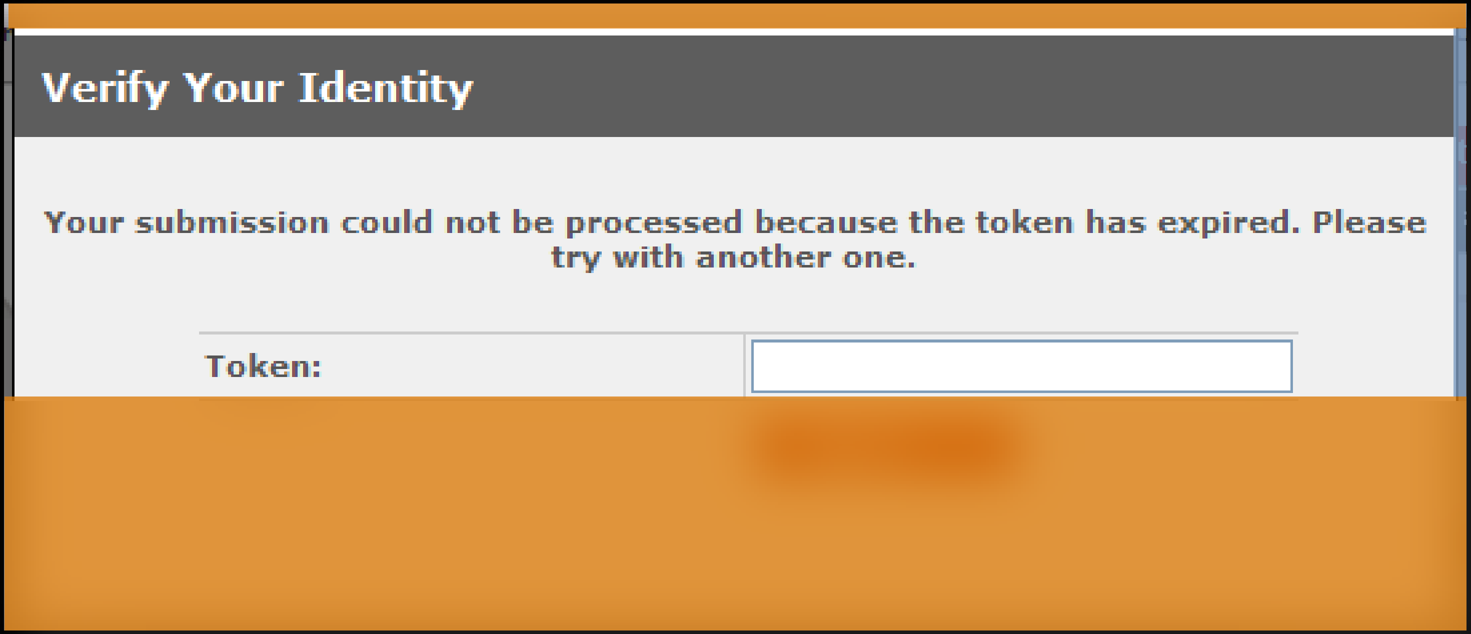


At the moment, is the process of gathering unique data on your system to create a unique digital signature (UDS). In the future the system will identify your computer by UDS. Please enter the following information:

Your telephone banking access code	
Your date of birth	dd / mm / yyyy

Sign In – Token

- Got a token for your corporate account? Do you still feel safe?

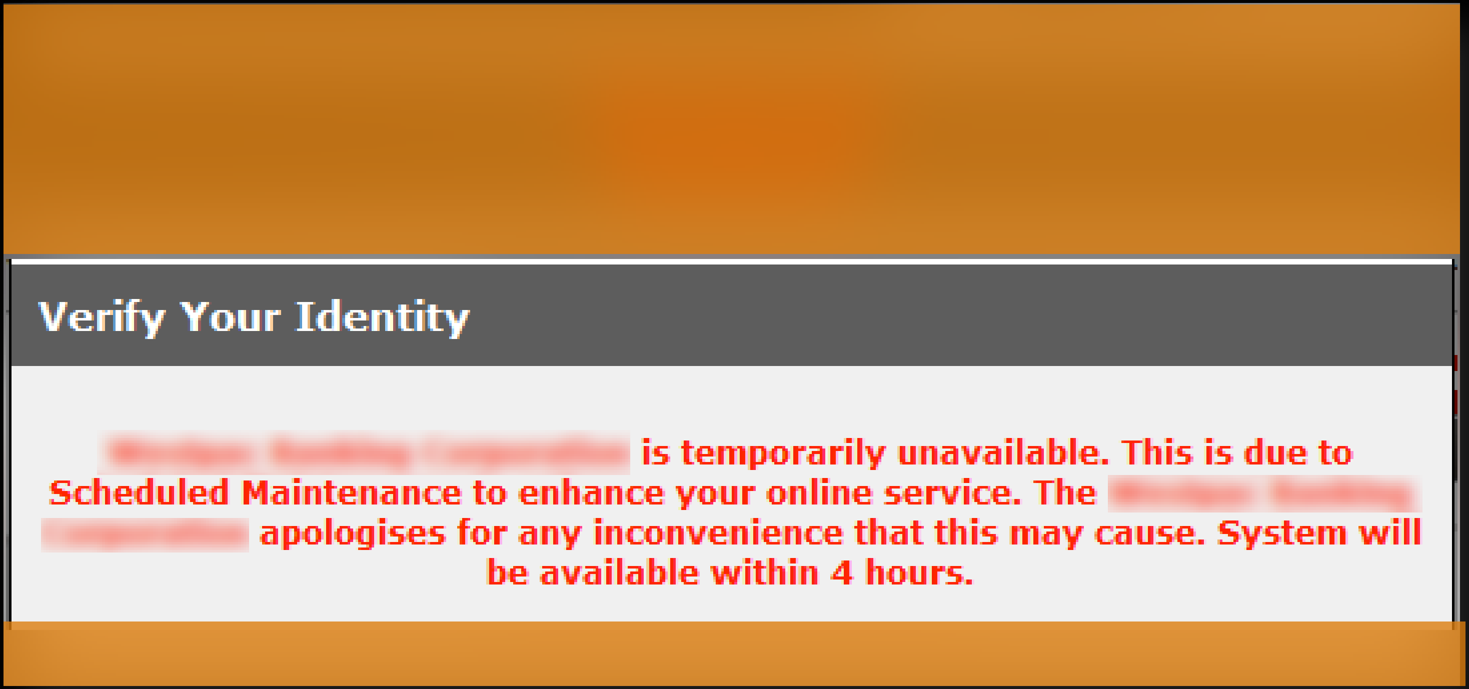


Your submission could not be processed because the token has expired. Please try with another one.

Token:	

Sign In – Token

- Now you are locked out while they buy enough time to transfer money

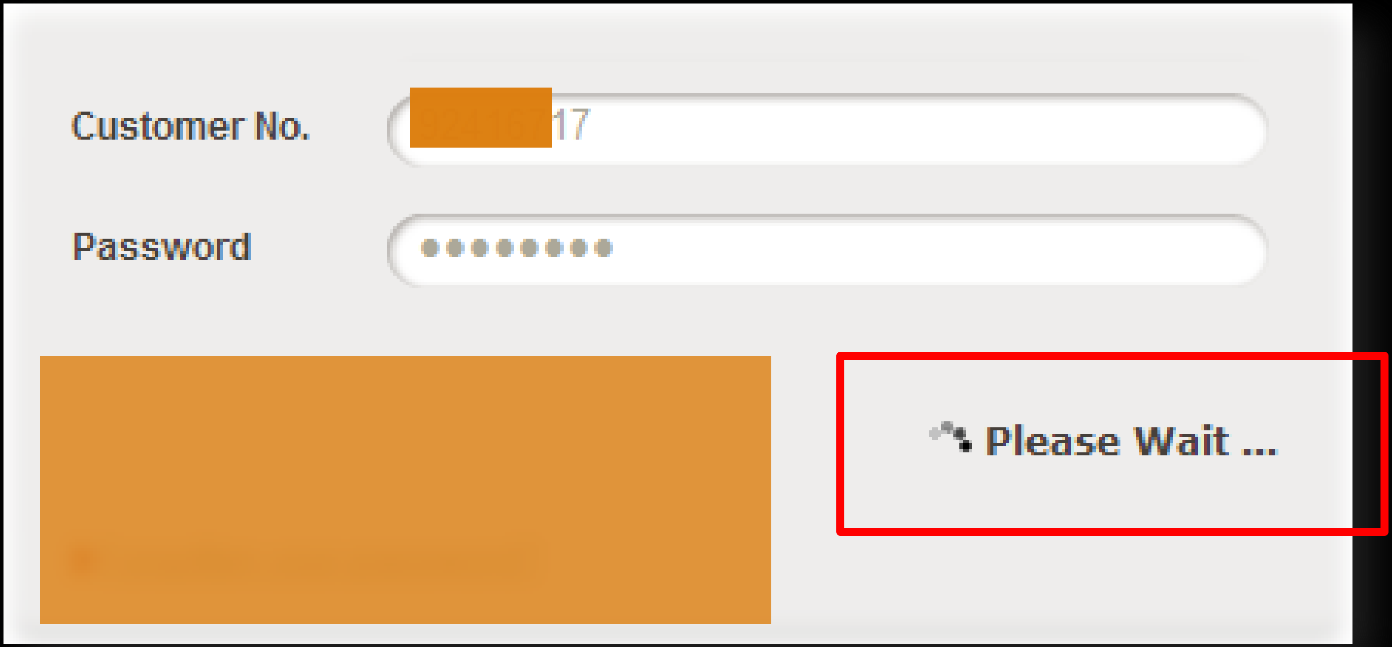


Verify Your Identity

[redacted] is temporarily unavailable. This is due to Scheduled Maintenance to enhance your online service. The [redacted] [redacted] apologises for any inconvenience that this may cause. System will be available within 4 hours.

Sign In - MITM

- There is no such a thing as ‘Please Wait’ in the online banking page.



Customer No. [redacted]17

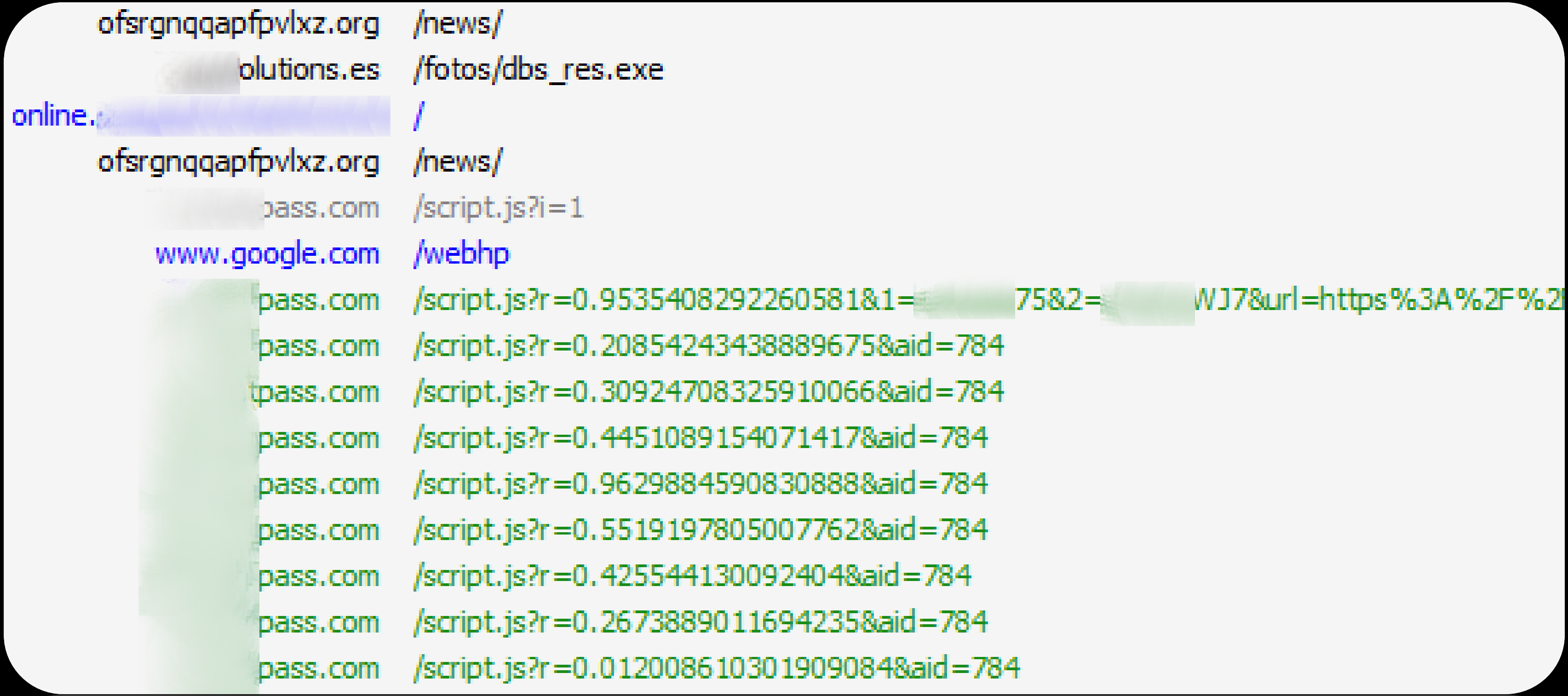
Password ●●●●●●●●●

[redacted]

Please Wait ...

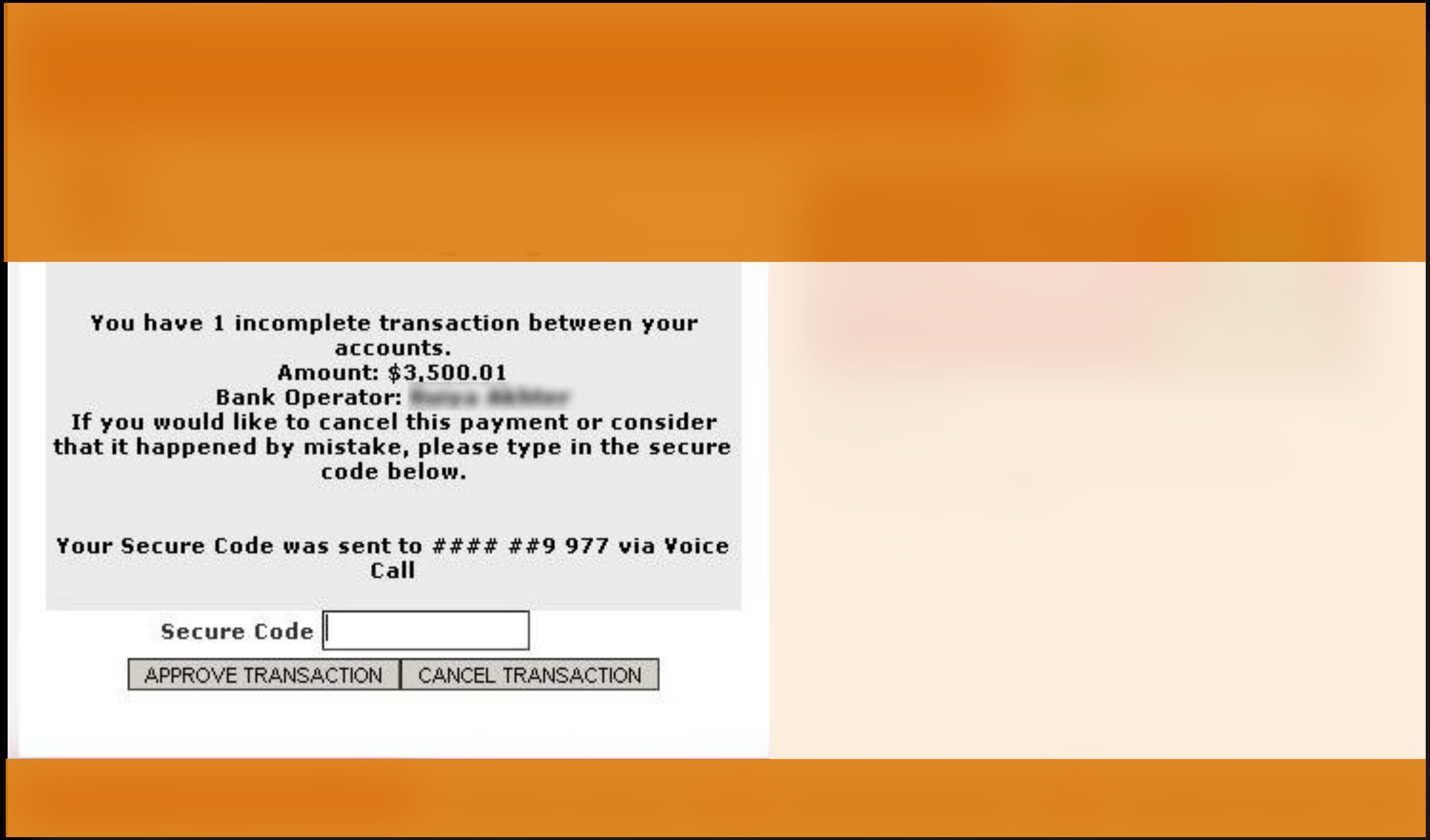
Sign In - MITM

- What's happening while you are waiting…



```
      ofsrgnqqapfpvlxz.org   /news/
      [redacted]olutions.es   /fotos/dbs_res.exe
online.[redacted]            /
      ofsrgnqqapfpvlxz.org   /news/
          [redacted]ass.com  /script.js?i=1
            www.google.com   /webhp
          [redacted]pass.com /script.js?r=0.9535408292260581&1=[redacted]75&2=[redacted]WJ7&url=https%3A%2F%2
          [redacted]pass.com /script.js?r=0.20854243438889675&aid=784
          [redacted]pass.com /script.js?r=0.30924708325910066&aid=784
          [redacted]pass.com /script.js?r=0.4451089154071417&aid=784
          [redacted]pass.com /script.js?r=0.9629884590830888&aid=784
          [redacted]pass.com /script.js?r=0.5519197805007762&aid=784
          [redacted]pass.com /script.js?r=0.425544130092404&aid=784
          [redacted]pass.com /script.js?r=0.2673889011694235&aid=784
          [redacted]pass.com /script.js?r=0.012008610301909084&aid=784
```

Transaction Injection - SMS



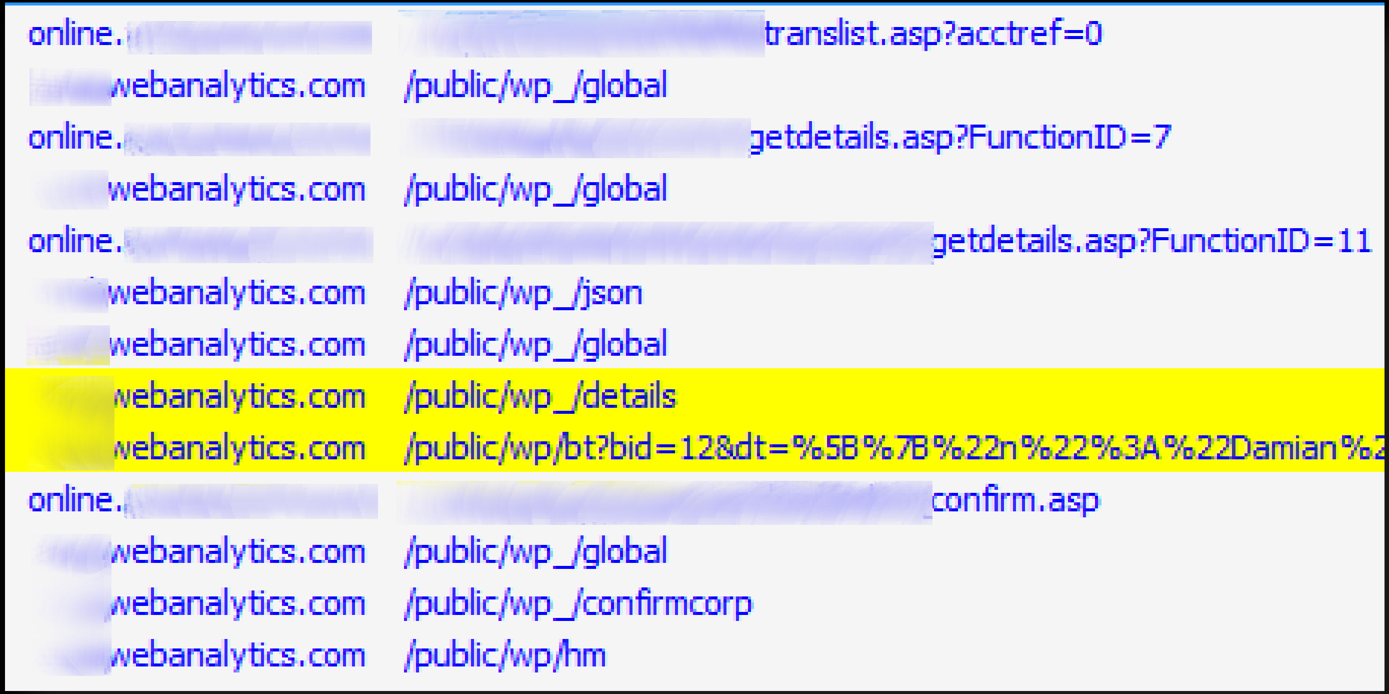
You have 1 incomplete transaction between your accounts.
Amount: $3,500.01
Bank Operator: [redacted]
If you would like to cancel this payment or consider that it happened by mistake, please type in the secure code below.

Your Secure Code was sent to #### ##9 977 via Voice Call

Secure Code

Transaction Manipulation

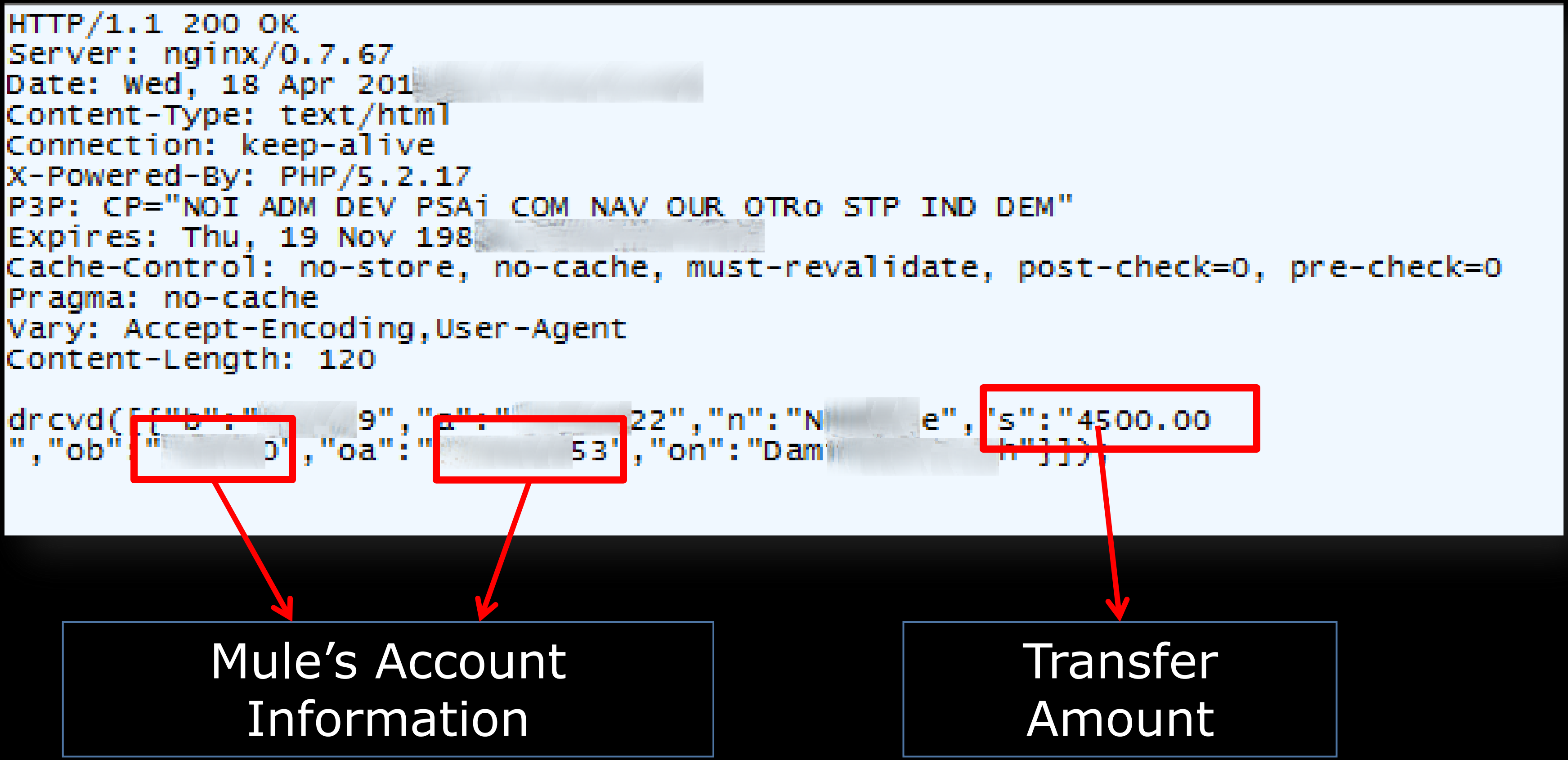
- Even when there is no visual sign of infection, it can happen silently.
- C&C communication during Tx pages



```
online.[redacted]        [redacted]translist.asp?acctref=0
[redacted]webanalytics.com   /public/wp_/global
online.[redacted]        [redacted]getdetails.asp?FunctionID=7
[redacted]webanalytics.com   /public/wp_/global
online.[redacted]        [redacted]getdetails.asp?FunctionID=11
[redacted]webanalytics.com   /public/wp_/json
[redacted]webanalytics.com   /public/wp_/global
[redacted]webanalytics.com   /public/wp_/details
[redacted]webanalytics.com   /public/wp/bt?bid=12&dt=%5B%7B%22n%22%3A%22Damian%2
online.[redacted]        [redacted]confirm.asp
[redacted]webanalytics.com   /public/wp_/global
[redacted]webanalytics.com   /public/wp_/confirmcorp
[redacted]webanalytics.com   /public/wp/hm
```

Transaction Manipulation

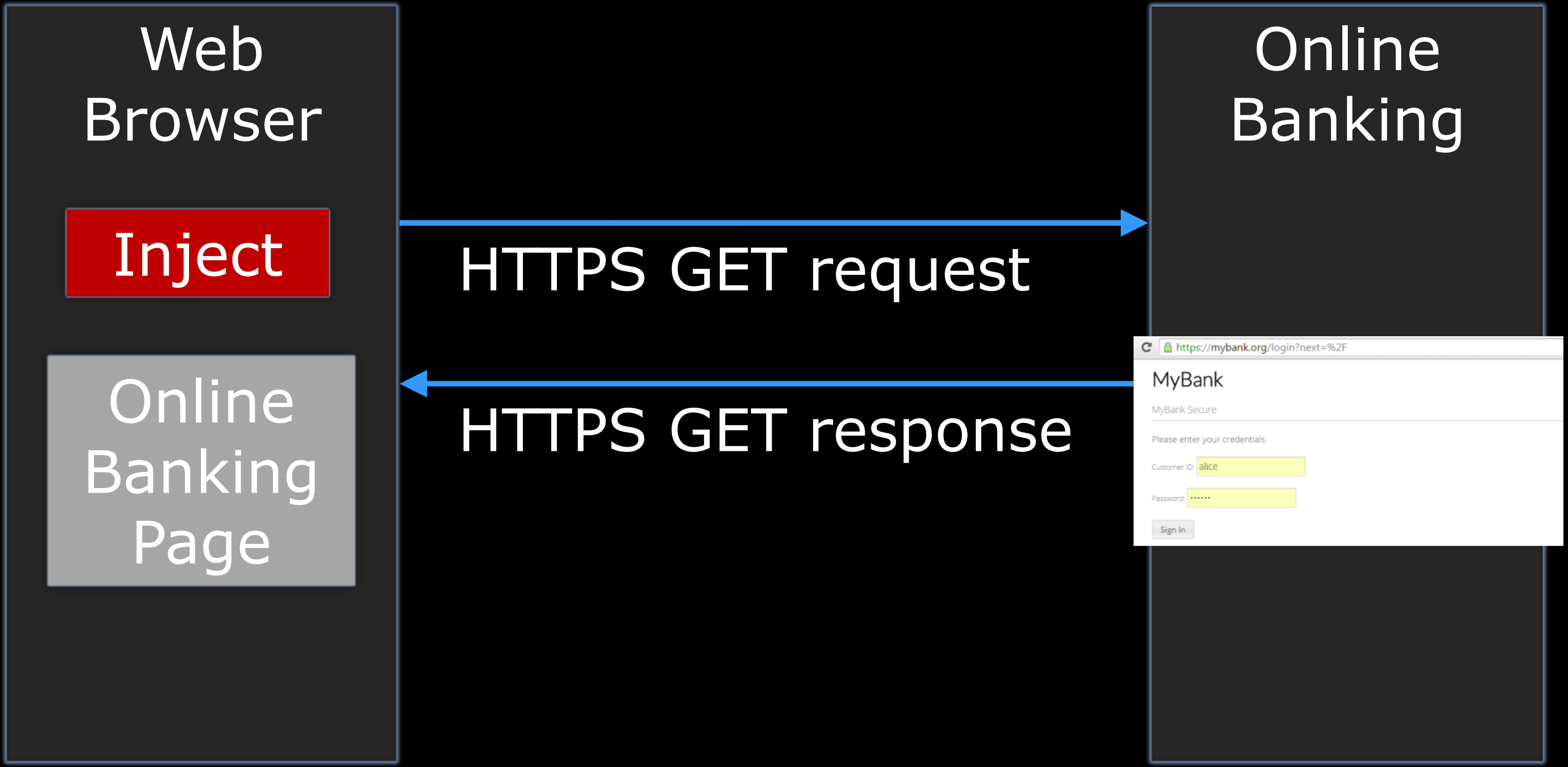
- What is the malware receiving? → Inject and Mule



```
HTTP/1.1 200 OK
Server: nginx/0.7.67
Date: Wed, 18 Apr 201[redacted]
Content-Type: text/html
Connection: keep-alive
X-Powered-By: PHP/5.2.17
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Expires: Thu, 19 Nov 198[redacted]
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding,User-Agent
Content-Length: 120

drcvd([{"b":"[redacted]9","a":"[redacted]22","n":"N[redacted]e","s":"4500.00
","ob":"[redacted]0","oa":"[redacted]53","on":"Dam[redacted]h"}]);
```

Basic Concept

DOM Injection



https://mybank.org/login?next=%2F

MyBank

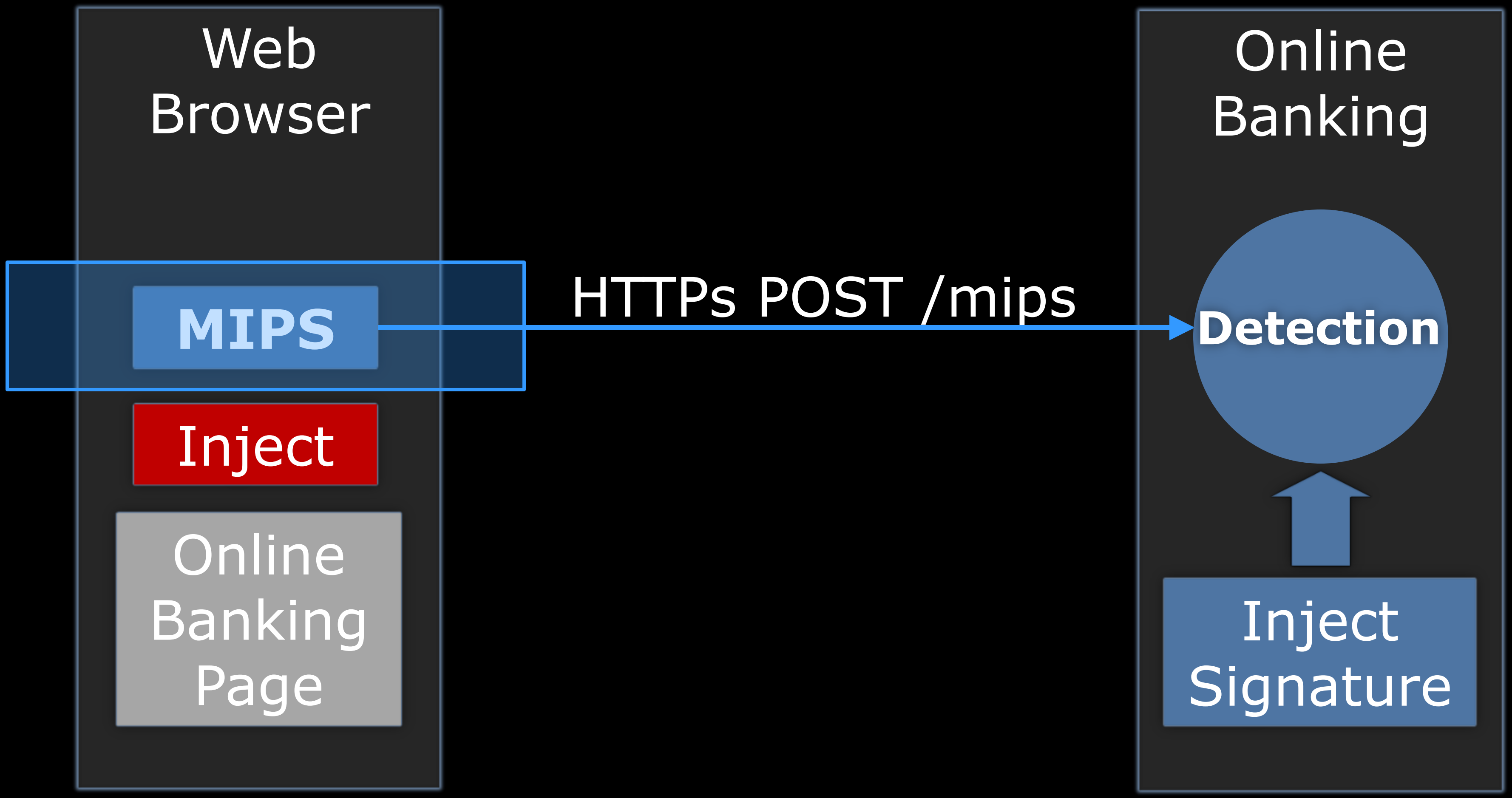
MyBank Secure

Please enter your credentials.

Customer ID: alice

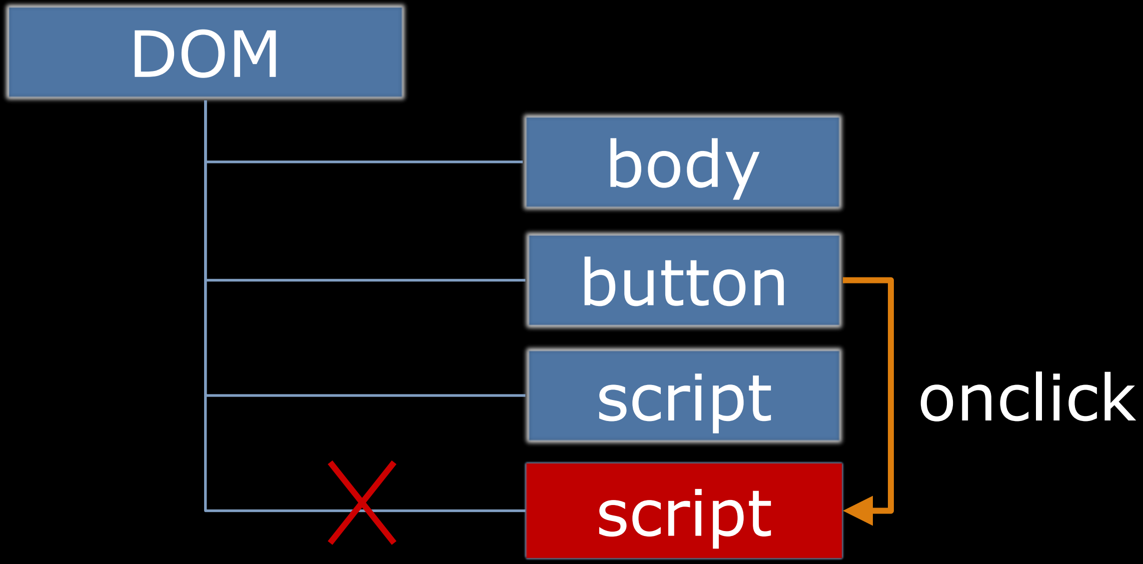
Password: ••••••

DOM Scan
: MIPS (**M**alware **I**nject **P**robe **S**ystem)



Attacks and Defenses

Attack: DOM Stealth



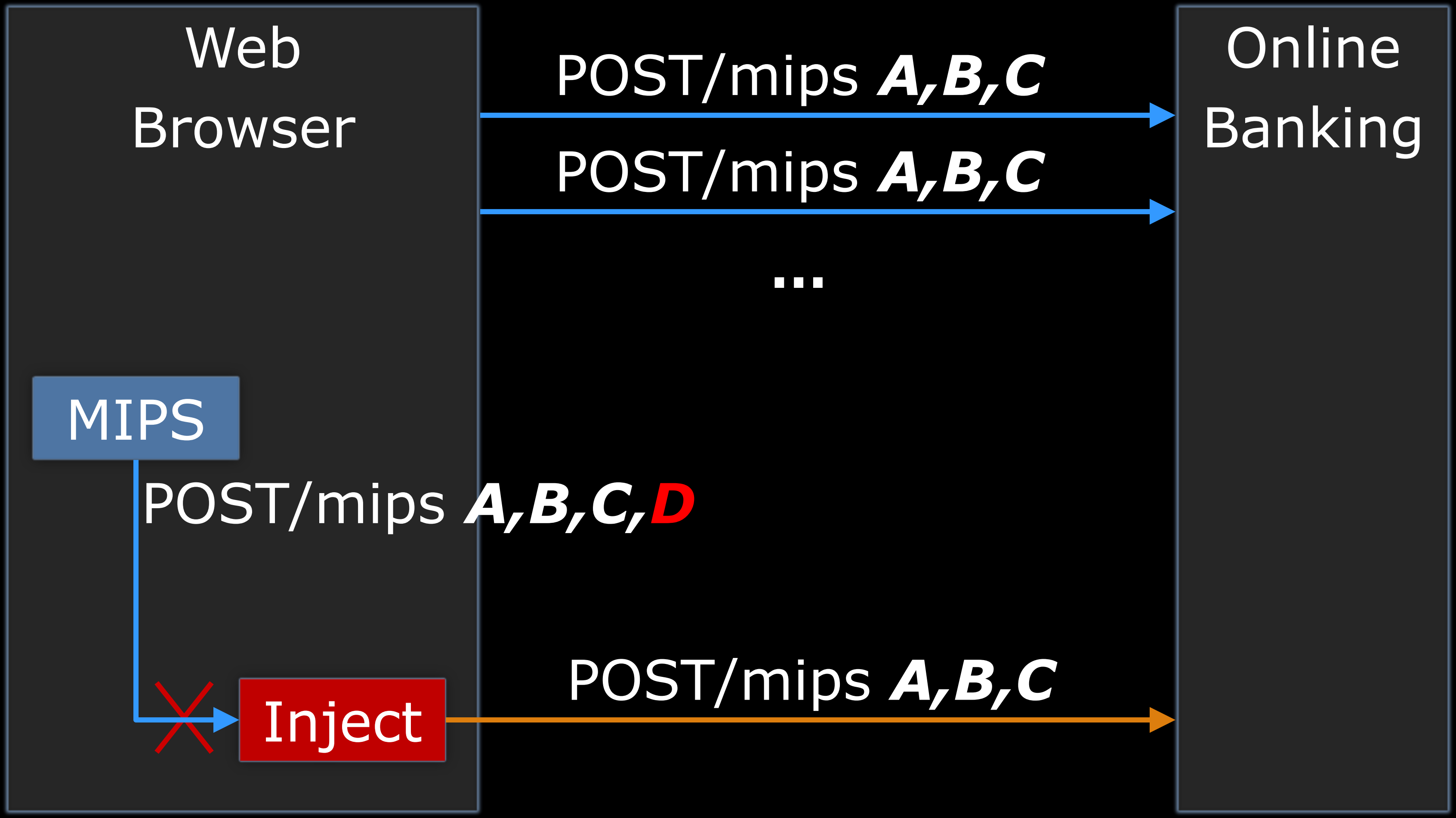
- Malware inject removes itself, but it still remains in the memory
- Exploit memory leak patterns
  - Dangling/circular references, closures

Defense: DOM Event Scan

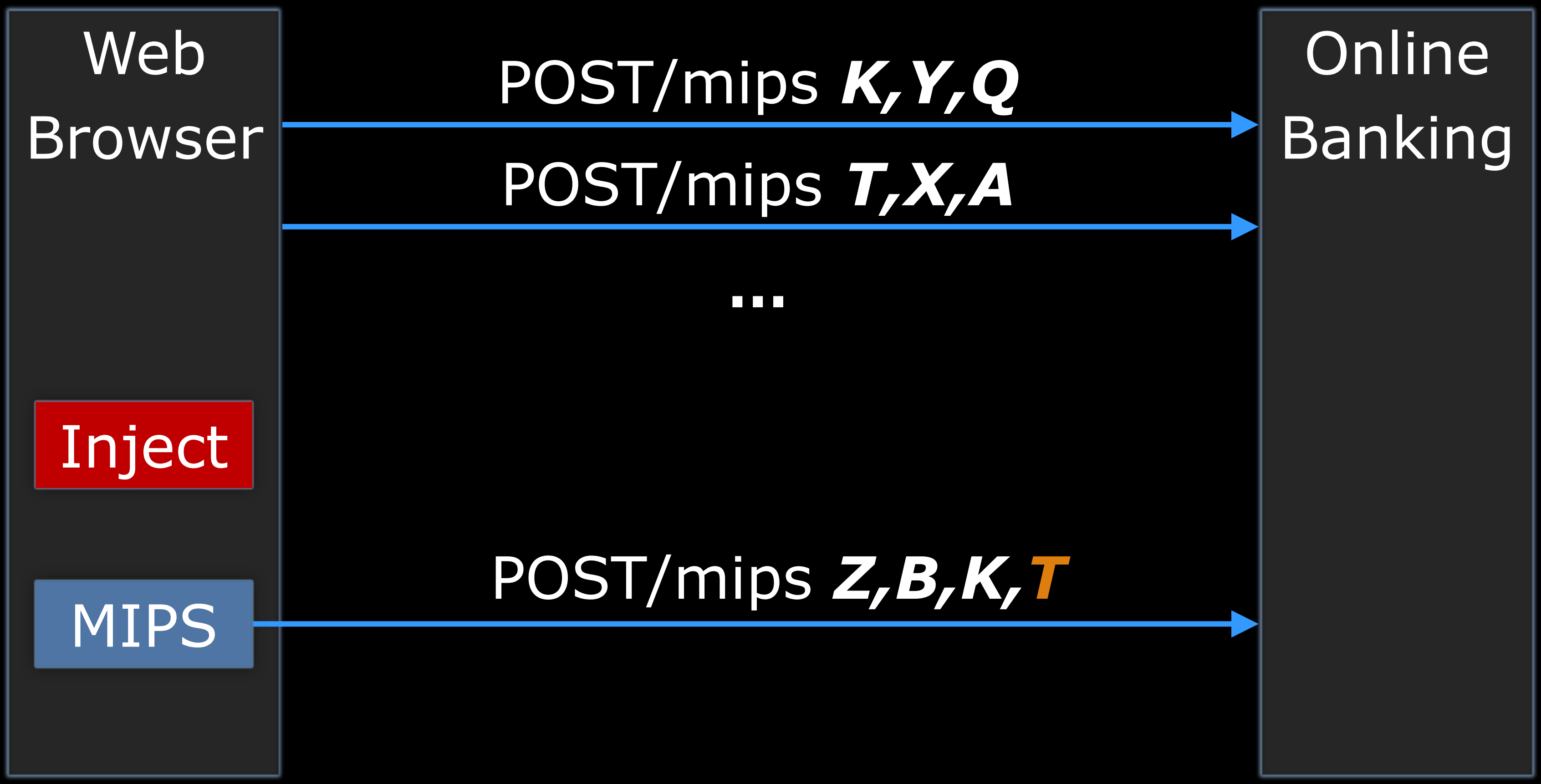
- Identify entry points (unload, click, timer)
- Enumerate event handlers

```
element.onclick = handler
   Scan: element.onclick
element.addEventListener
   Scan: getEventListeners(element, “click”)
$(element).on(“click”, handler)
   Scan: $._data(element, "events" )
$(element).observe(“click”, handler)
   Scan: element.getStorage().
   get('prototype_event_registry').get('click')
```

Attack: Replay

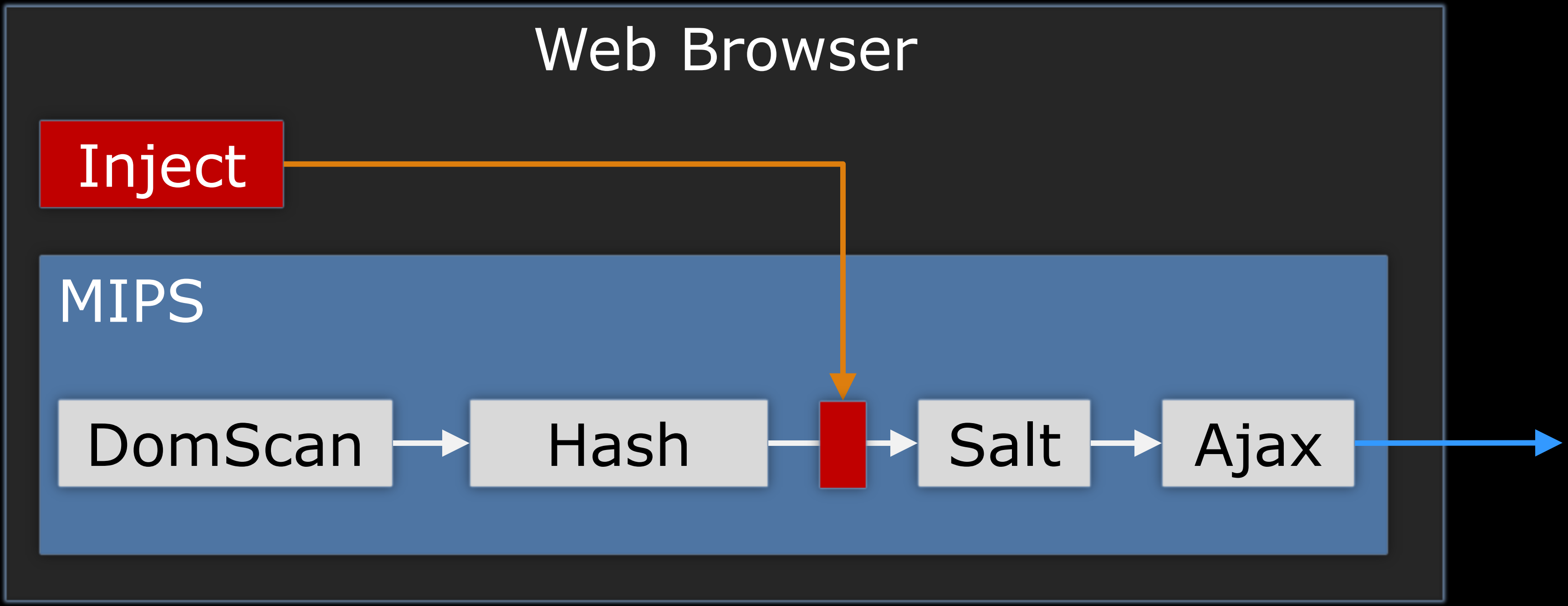


Defense: Salting



- Original MIPS intel gets transformed differently each time using the random variable

Attack: Forging MIPS Intel



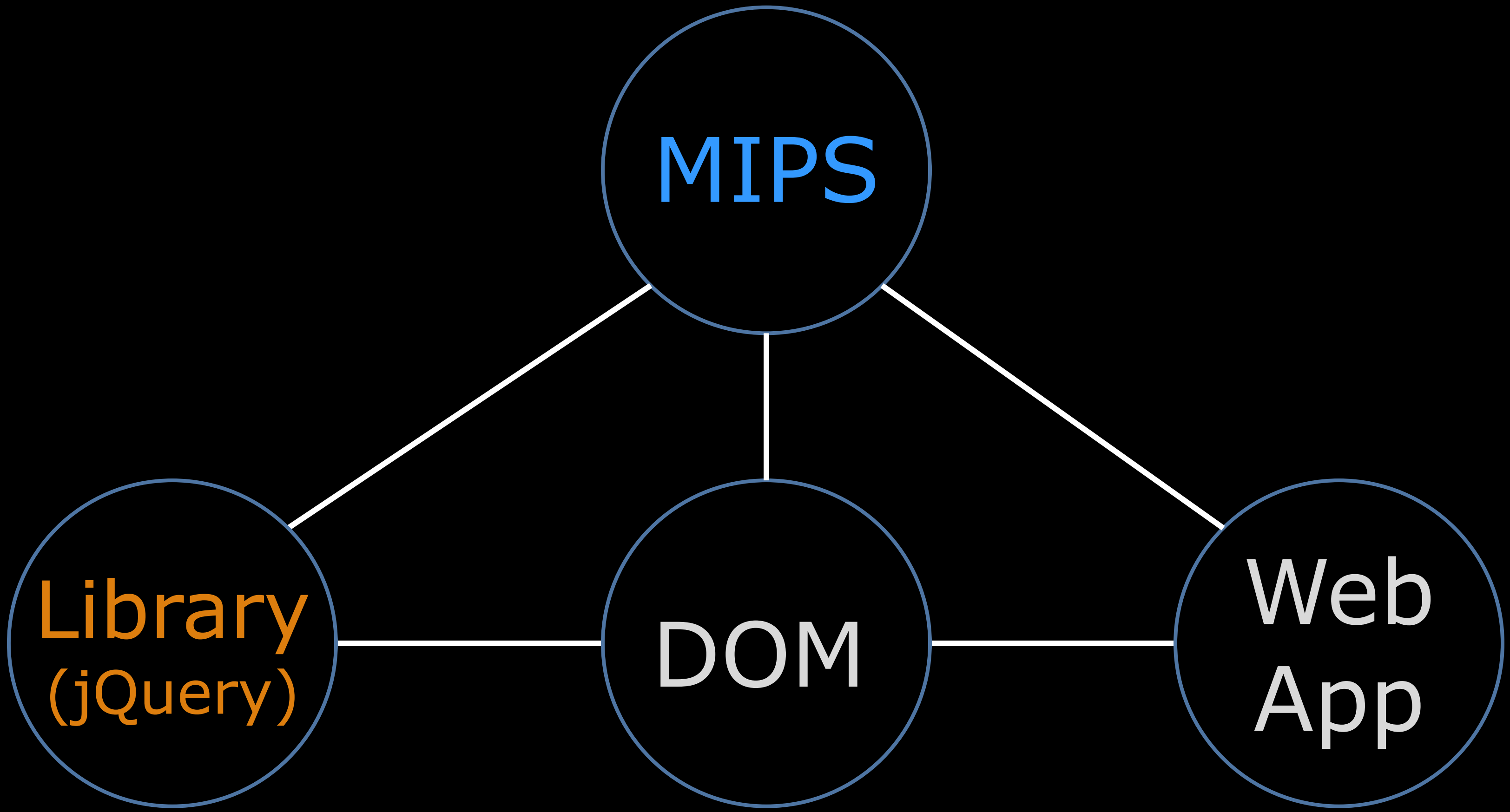
- Hook Salt() and modify hashes OR
- Block MIPS & call MIPS functions as necessary

Defense: Code Integrity Check

- Call stack context

```
var Check = function(na, nb) {
  var SecureCheck = function(na, nb) {
    var callee = na ^ crc32(arguments.callee);
    var caller = nb ^ crc32(arguments.callee.caller);
    return callee ^ caller ^ DomCheck();
  };
  return SecureCheck(na, nb);
};

var na = 32053221, nb = 4321053;
result = Check(na, nb);
```

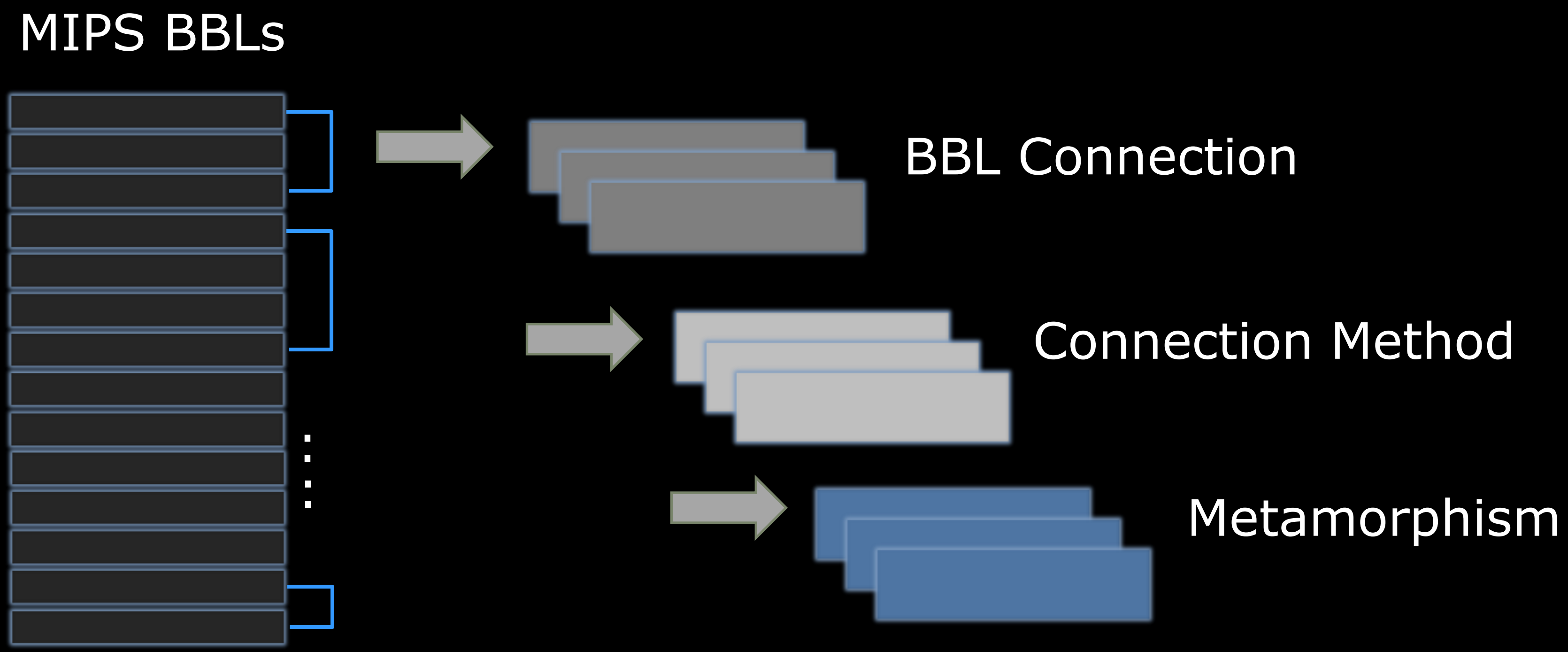
Defense: Code Integrity Check



Defense: Randomisation

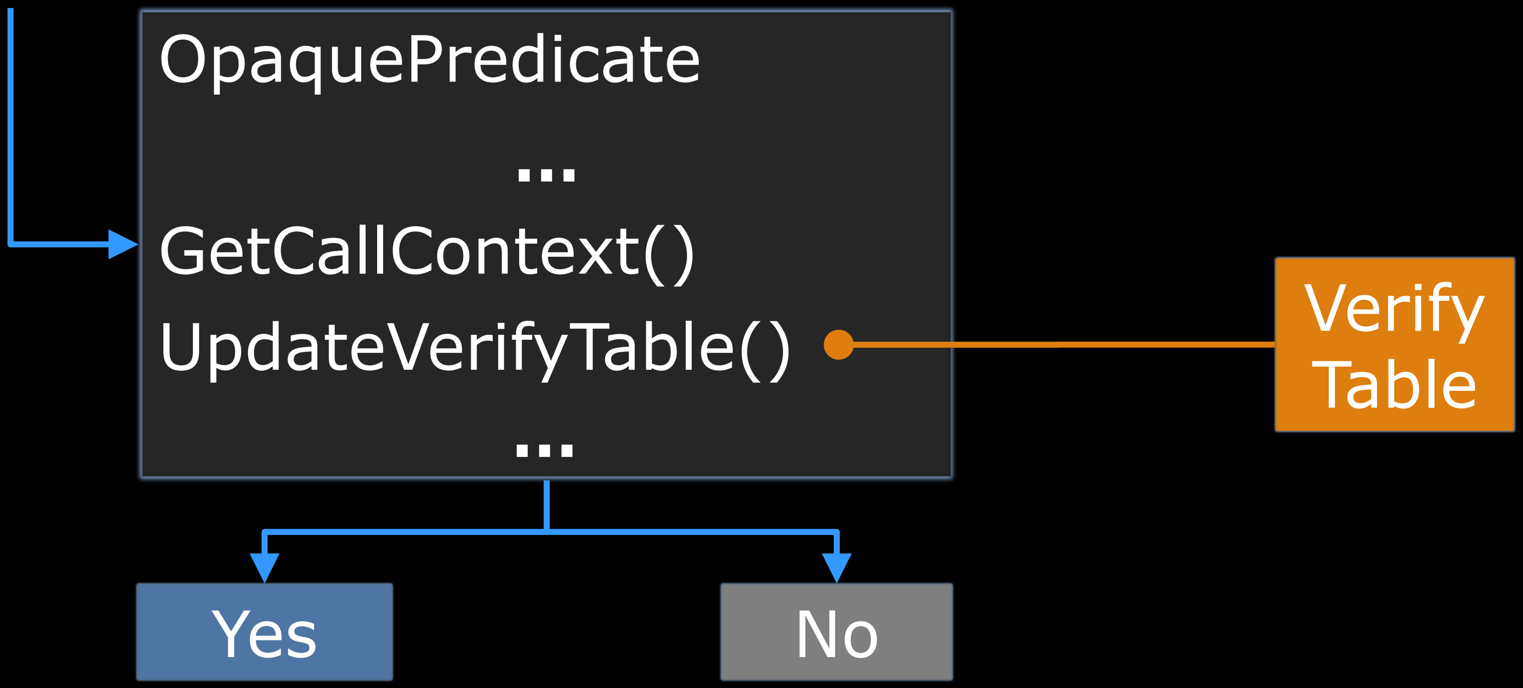
- Problem with integrity check
  - Malware Regexes, modifies and reconstruct MIPS
  - Malware simulates MIPS with bypass code
- Strategy
  - Polymorphism
  - Maintain a set of ***algorithmically heterogeneous*** MIPS code
  - Fragmented random MIPS scripts with different names

Defense:
Control Flow Randomisation



- Chain of Randomisations starting with basic blocks (BBLs) of MIPS code

Defense: Opaque Predicates



- Retrieve call context in deeply buried OP
- Insert part of main logic within OP

Attack: Rootkit

```
var original_func = document.getElementsByTagName;
document.getElementsByTagName = function () {
  r = original_func.apply(document, arguments);
  for(var i=0; i<r.length; i++) {
    var inject_signature = ‘string_in_my_inject';
    if(r[i].text.search(inject_signature) != -1) {
      r[i].remove();
      console.log('Inject Rootkitted!');
      break;
    }
  }
  return r;
};
```

Defense: Detecting Rootkits

- Deliberately trigger exception → Call stack

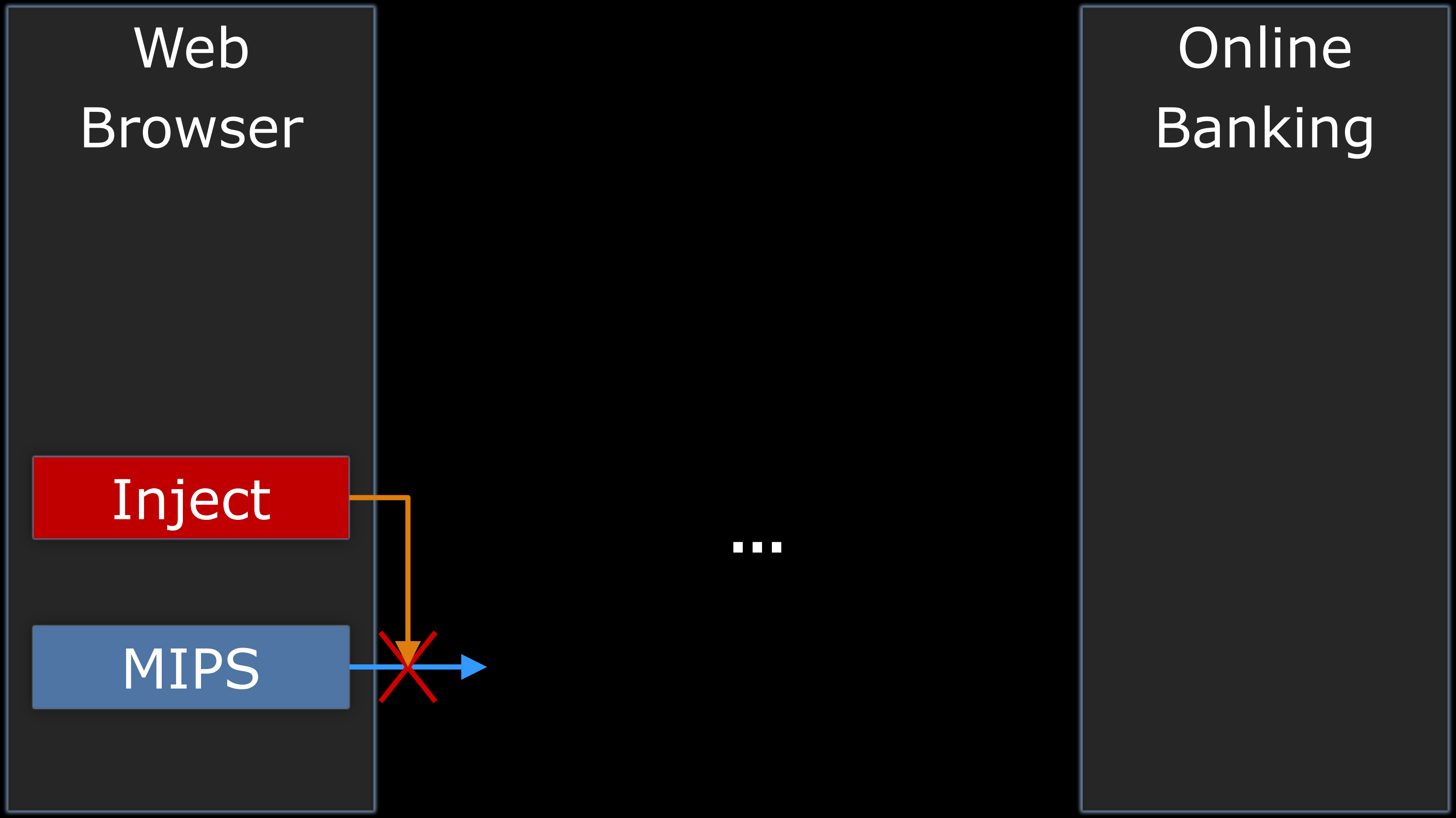
```
var hooked = Function.prototype.toString;
Function.prototype.toString = function() {
      hooked.apply(this, arguments);
} // DOM Rootkit

var TriggerException = function(){
    try {
        Function.prototype.toString.call('hooktest')
    }
    catch(err) {
        console.log(err.stack);
    }
}
TriggerException();
```

Defense: Detecting Rootkits

- Is the red line present in a clean session?

```
TypeError: Function.prototype.toString is not
generic
  at String.toString (native)
  at String.Function.toString(/login?next=%2F:173:7)
  at TriggerException (/login?next=%2F:177:29)
  at https://mybank.org/login?next=%2F:183:1
```

Attack: Blocking MIPS



Defense: MISSING_MIPS Event

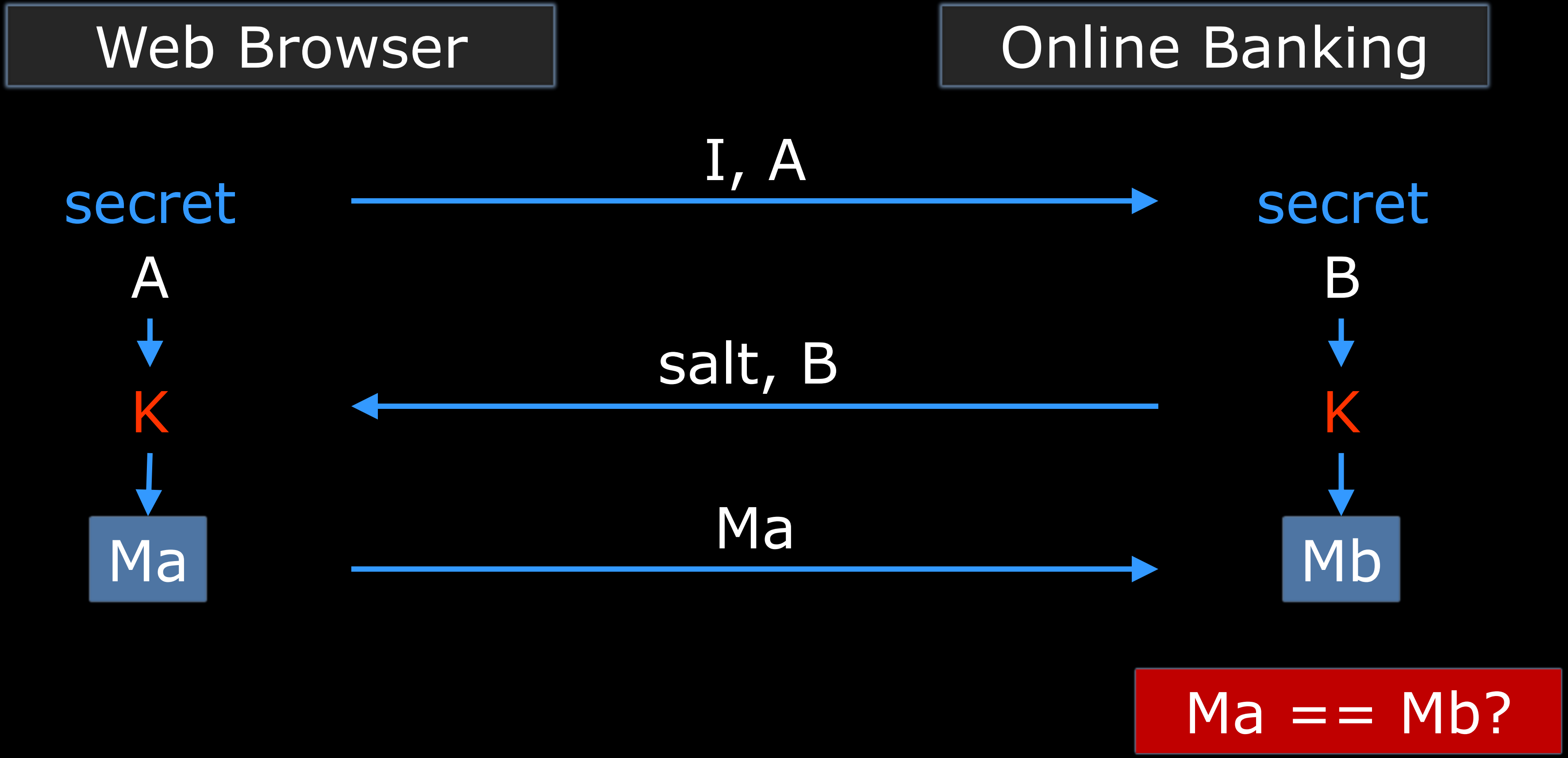
- MISSING-MIPS Event should be implemented on the online banking server side if MIPS is not the integral part of online banking logic
- Method
  - Ensure MIPS intel is not cached by the proxy in-between
  - Correlate web access log with MIPS log

Detecting Moving Targets

- Detect evolving injects
  - Effective on minor inject upgrade
- Methods
  - Locality sensitive hashing (i.e. TLSH)

ZKP: SRP (By Tom Wu, Stanford)

- Over-simplified Secure Remote Password



Use Cases

- MITM attack
  - No shared secrets get transmitted on the wire (password, OTP code)
- Passive sniffing
  - Force attackers to place injects (so we can detect it!)
- MIPS hardening
  - DOM function integrity data
  - MIPS integrity data
  - MIPS rootkit detection data
  - MIPS intelligence format

Conclusion

- Diversity of implementation is the key for survival
- Be creative and out-smart the cybercriminals!
- Perform application security check

Thank You

Sean Park
Senior Malware Scientist
TrendMicro