CISPA Center for IT-Security, Privacy and Accountability

UNIVERSITÄT DES SAARLANDES

# Schutz des Online-Banking-Browsers

## BOB-Symposium

Prof. Dr. Christian Rossow
Michael Brengel

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

max planck institut informatik

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Max Planck Institute for Software Systems

# Der Browser beim Online-Banking

- Browser als Basis für Online-Banking
  - Großteil der Online-Banking-Geschäfte werden mit Browsern abgewickelt
  - Kleiner Markt der Browser (Chrome, Internet Explorer + Firefox = 90%)


# Zielsetzung: Schutz des Browsers

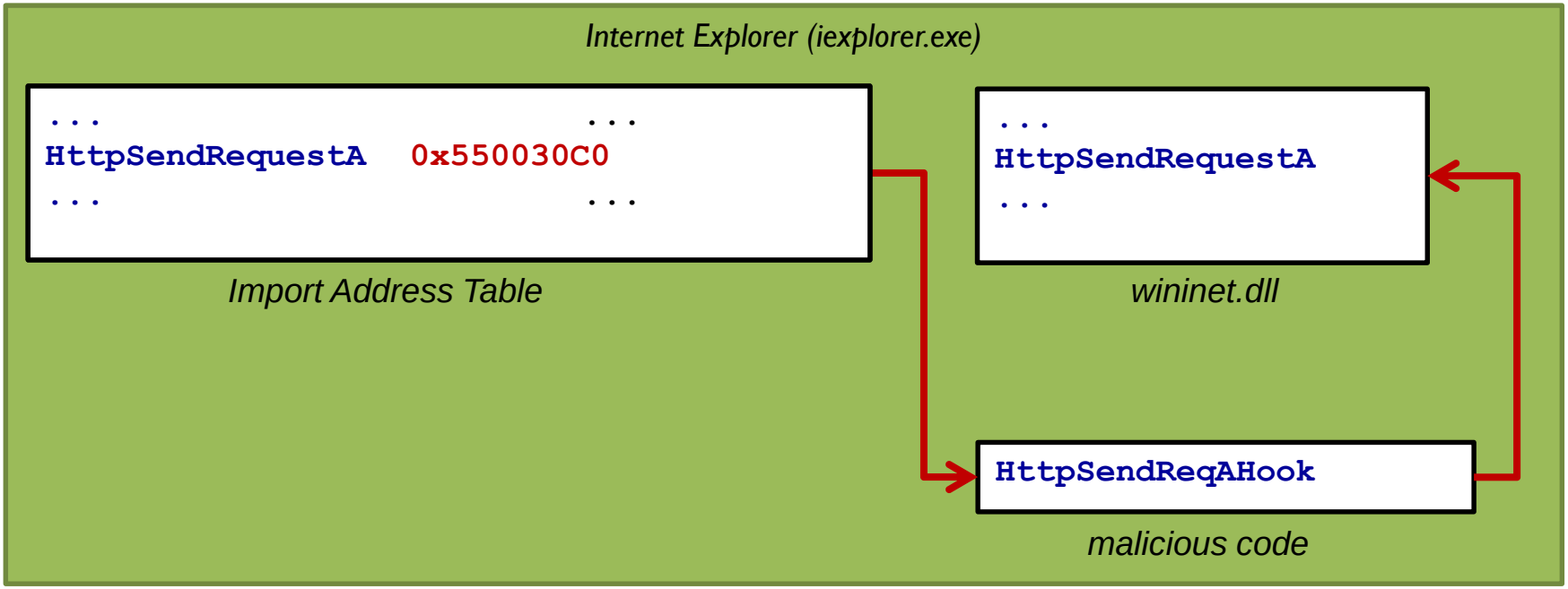
- Teilprojekt der Universität des Saarlandes
  - Ziel A: Verständnis über potentielle Angriffe auf den Browser
  - Ziel B: Verständnis über Sicherheit von Mobile Banking Apps
  - Ziel C: Browser vor Angriffen schützen
  - Ziel D: Clientseitige Manipulationserkennung

# Ziel A: Verständnis über Angriffe

- Analyse bekannter Banking-Trojaner
  - Citadel
  - Carberp
  - Vawtrak
  - Dridex
  - Tinba
  - Dyre

# Ziel A: Verständnis über Angriffe

- Banking-Trojaner manipulieren Browser über Hooking
  - Typische Funktionsaufrufe werden auf Schadcode umgelenkt
  - Trojaner kann so bspw. Netzwerkverkehr mitlesen und manipulieren



# Ziel A: Verständnis über Angriffe

- Dynamische Manipulation von Web-Inhalten („Web Injects“)

```
Target Pattern(s):
(?:^https://banking\.postbank\.de/app/legitimation)
```

**Welche URL?**

```
pcre_pattern
(?:</form>(?P<inject>))
data_end
```

**Wo einbinden?**

```
data_inject
<p class="pBlock center"><strong>%2</strong></p>
<select name="iTAN" style="display: none;">
  <option value="%1" selected="selected"></option>
</select>
data_end
```

**Was injizieren?**

# Ziel B: Sicherheit von Mobile Banking Apps

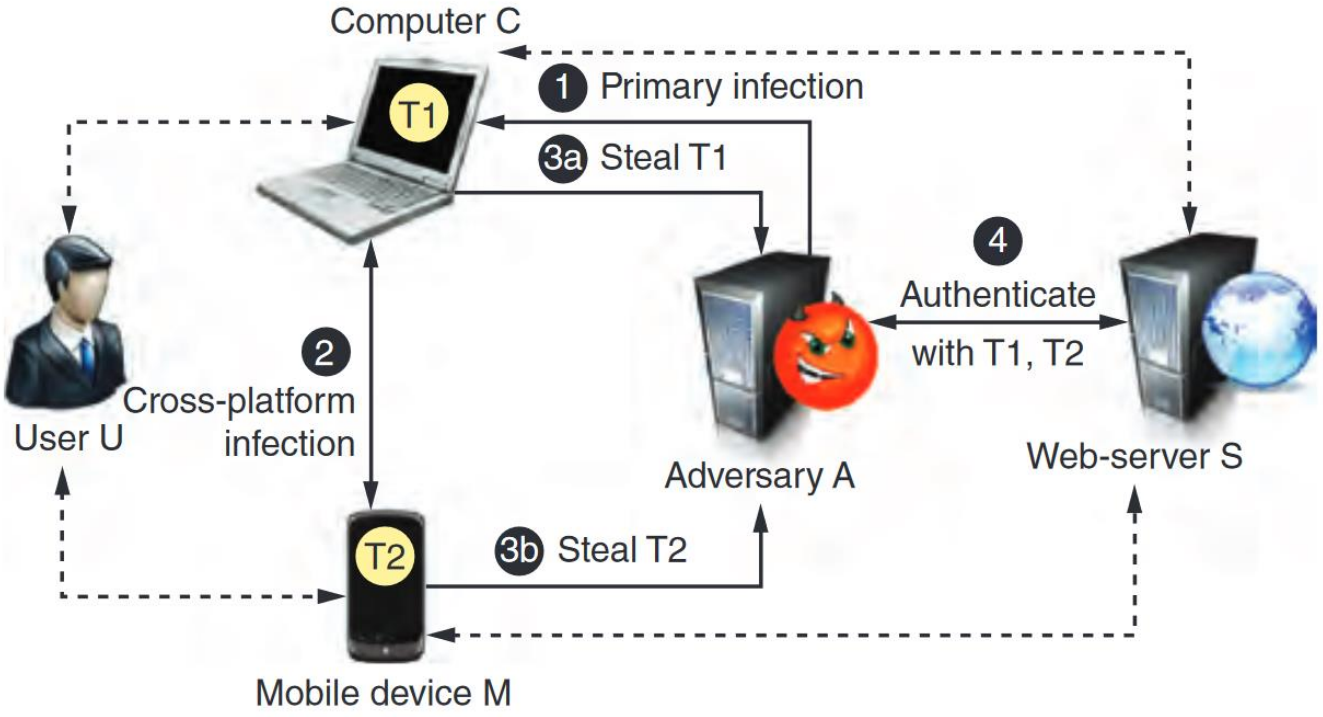
CISPA Center for IT-Security, Privacy and Accountability

- Analyse bestehender Banking-Apps der größten 8 Banken in DE
  - Größtenteils sichere Implementierungen
  - 3 von 10 Apps obfuskieren den Code (Anti-Reversing)
  - 10 von 10 Apps schützen vor Man-in-the-Middle-Angriffen
  - 4 von 10 Apps haben teils unverschlüsselte Kommunikation via HTTP
  - 1 von 10 Apps schützen gegen Repacking

- Darüber hinausgehende Empfehlungen:
  - Daten (Cache, etc.) App-lokal und verschlüsselt speichern
  - Strikte Trennung Banking-Gerät und 2FA-Gerät

# Ziel B: Sicherheit von Mobile Banking Apps

- Angriffe möglich, wenn Smartphone als 2FA eingesetzt
  - Trojaner auf dem PC infiziert installiert Malware auf Smartphone
  - Smartphone-basierte TAN-Verfahren können dann manipuliert werden (mobileTAN, photoTAN, etc.)



# Ziel C: Schutz des Browsers

- Ist ein sicherer Browser möglich?

- Potentielle Schutzmaßnahmen
  - Verschleierung des Programmcodes
  - Statisches Linken verwendeter Systembibliotheken
  - Stetige Integritätsprüfungen
  - Striktes Pinning des Browsers an eine dedizierte Bank
  - Anti-Debugging und Anti-Keylogging-Mechanismen

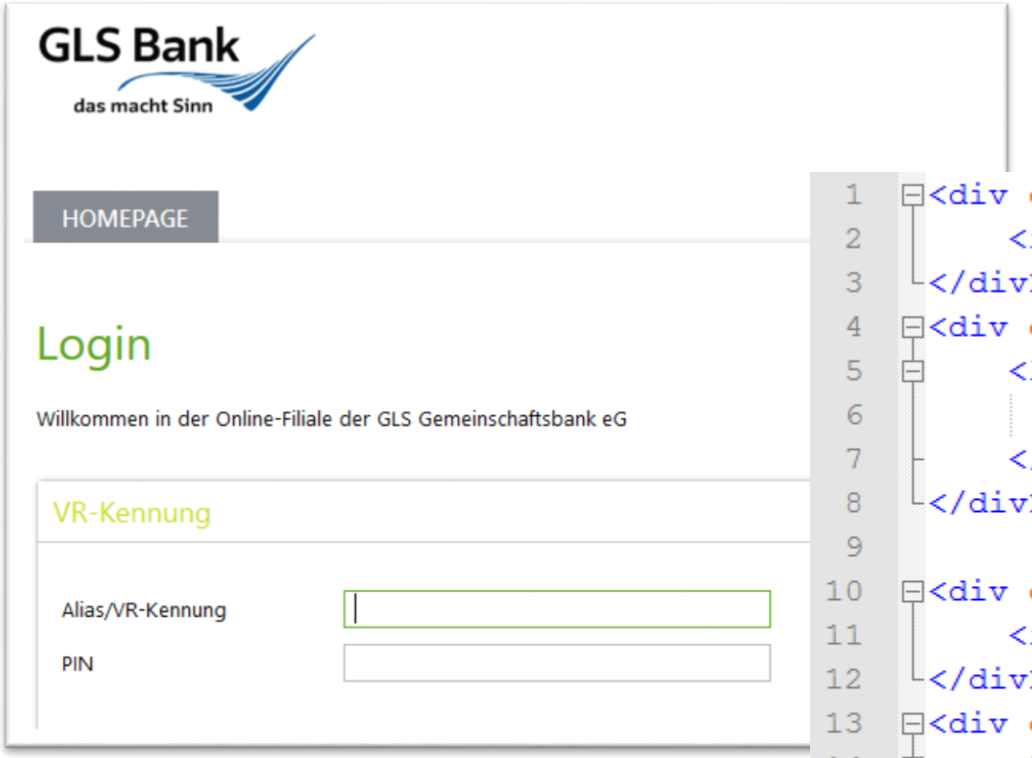
# Ziel C: Schutz des Browsers

- Sicherheitsanalyse eines gehärteten Browsers
  - Zusammenarbeit mit Bank und Browser-Hersteller
  - Pentesting und Identifikation von mehreren Schwachstellen
  - Vorschlag von Verbesserungsmaßnahmen
  - Neue Version mit erweiterten Schutzmaßnahmen angedacht

- Generelles Fazit:
  - Gänzlicher Ausschluss eines Angriffs ist unmöglich, da der Angreifer mit seiner Malware “zu stark” ist (volle Kontrolle über System)
  - Aber: Aufwand des Angreifers wird durch Browser **deutlich** erhöht

# Ziel D: Clientseitige Manipulationserkennung

- Aufbau einer Banking-Webseite (Beispiel Fiducia & GAD)




```
1  <div class="gad-decoratedControl ym-g70">
2      <input maxlength="35" id="vrkennungalias" ..
3  </div>
4  <div class="gad-blockHeader ym-g30 ym-gl">
5      <label for="vrkennungalias" class="gad-label
6          Alias&#x2f;VR-Kennung
7      </label>
8  </div>
9
10 <div class="gad-decoratedControl ym-g70">
11     <input maxlength="20" id="pin" ... />
12 </div>
13 <div class="gad-blockHeader ym-g30 ym-gl">
14     <label for="pin" class="gad-label">
15         PIN
16     </label>
17 </div>
```

# Ziel D: Clientseitige Manipulationserkennung

- Prototyp: Erkennung von Veränderungen im HTML-Gerüst
  - XML xPath-Expressions definieren sensitive Codebereiche
  - “Signatur” über **im Browser angezeigten** Code wird Bank zugesendet
  - Bank berechnet ihrerseits Signatur über **ausgelieferten** Code
  - Prüfung durch Bank: `sign(angezeigt) =?= sign(ausgeliefert)`

# Ziel D: Clientseitige Manipulationserkennung

- Herausforderungen
  - Dynamische Webseitenelemente (z. B. Kalender) ignorieren
  - Browser haben unterschiedliche Repräsentationen des HTML-Gerüsts

- Evaluation des Prototyps
  - Änderung von Transaktionsformularinhalten wird erkannt ✓
  - Manipulation in Umsatzanzeige wird erkennt ✓
  - Hinzufügen weitere Login-Formularfelder wird erkannt ✓

# Fazit

- Browser ist Kernangriffspunkt von Banking-Trojanern
  - Banking-Trojaner greifen die gängigsten Browser an
  - Hooking und dynamische Webinjects erlauben Manipulationen

- Erforschte Schutzmaßnahmen
  - Gehärtete Browser erschweren die Angriffe deutlich
  - Empfehlungskatalog für die Entwicklung von Banking-Apps
  - Browser für JavaScript-seitige Integritätsprüfungen verwenden

CISPA Center for IT-Security, Privacy and Accountability

UNIVERSITÄT DES SAARLANDES

# Schutz des Online-Banking-Browsers

## BOB-Symposium

Prof. Dr. Christian Rossow
Michael Brengel

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

max planck institut informatik

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Max Planck Institute for Software Systems