

SMTP-Tunnel

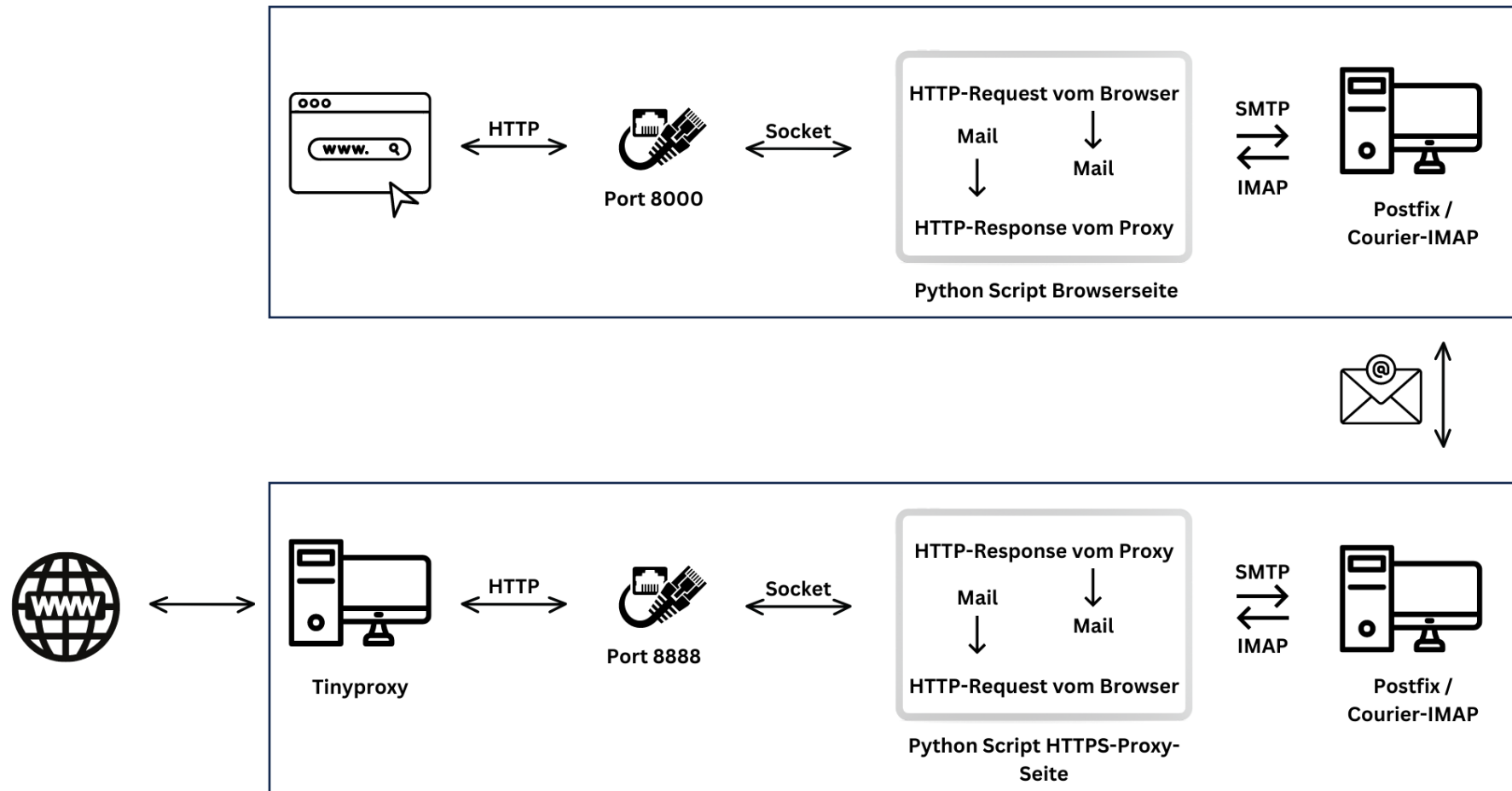
Ausarbeitung von Richard Herzog und Vanessa Lubrich



Gliederung

- Grundlegender Aufbau
- Demonstration
- Technische Besonderheiten
- Schutzpotenzial

Grundlegender Aufbau



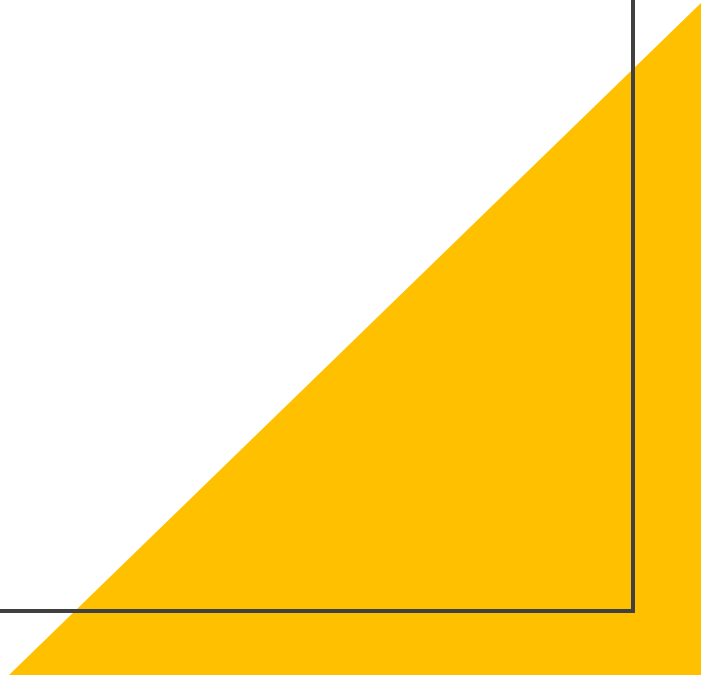
Demonstration



Technische Besonderheiten

Verwendete Tools

- Firefox Browser
- Tinyproxy
- Postfix
- Courier-IMAP



Postfix Konfiguration

- SMTP Tunnel mit Hilfe von Relay Hosts:

```
smtpd_relay_restrictions = permit_m  
myhostname = thnmlss-VirtualBox.fri  
alias_maps = hash:/etc/aliases  
alias_database = hash:/etc/aliases  
mydestination = $myhostname, thnmls  
relayhost = [10.0.2.4]  
mynetworks = 127.0.0.0/8 [::ffff:12  
mailbox_size_limit = 0
```

Python Scripts - Ablauf Browser Seite

Master Thread

- Kommandozeilenargumente auswerten
- Dem Browser einen Socket zur Verfügung stellen
- Ein einziges IMAP-Server-Objekt erstellen
- Neue Verbindungen vom Browser koordinieren
 - Browser-Request aus neuem Socket empfangen
 - Neues SMTP-Objekt für einen neuen Thread
 - Verbindung von Proxy Seite bestätigen lassen
 - Byte-Daten an Proxy Seite senden
 - Neuen Thread für Kommunikation starten

Neuer Thread vom Browser

- Warten auf Mails von der Proxy Seite:
 - Proxy benötigt mehr Daten -> Abfrage vom Browser Socket
 - Proxy sendet Daten -> Alle Daten abwarten und an Browser Socket übergeben
 - Proxy beendet Verbindung -> Verbindung wird geschlossen
- Senden von Browser-Daten an die Proxy Seite
 - Empfangen vom Browser Socket
 - Nichtleere Daten vom Socket erhalten -> Byte-Daten per Mail an Proxy Seite senden
 - Leeres Byte vom Browser -> Verbindung wird geschlossen -> Proxy wird nicht informiert
 - Timeout beim Empfangen vom Socket
 - Proxy hat Verbindung beendet -> Verbindung wird geschlossen
 - Proxy hat Verbindung nicht beendet -> Empfang vom Socket wird wiederholt, diesmal mit hohem Timeout-Wert
 - Timeout beim Empfangen vom Socket, oder leeres Byte empfangen -> Verbindung wird geschlossen -> Proxy wird nicht informiert
 - Nichtleere Daten vom Socket erhalten -> Byte-Daten per Mail an Proxy Seite senden

Python Scripts - Ablauf Proxy Seite

Master Thread

- Kommandozeilenargumente auswerten
- Ein einziges IMAP-Server-Objekt erstellen
- Verbindungsversuch von Browser Seite koordinieren
 - SMTP-Server Objekt für Verbindung erstellen
 - Bestätigungs-Mail mit Verbindungs-ID an Browser Seite senden
 - Neuen Socket für die Verbindung zum Proxy erstellen
 - Neuen Thread für Kommunikation starten

Erfolgreich hergestellte Verbindung

- Warten auf Mails von der Browser Seite
 - Daten aus Mails extrahieren und an den Proxy Socket übergeben
 - Wird länger keine Mail empfangen -> Verbindung wird geschlossen
- Antwort vom Proxy abwarten
 - Vollständigen Datenstrom vom Proxy Socket empfangen und per Mail an Browser Seite senden
 - Proxy sendet ein leeres Byte -> Verbindung wird geschlossen, Browser Seite wird informiert
 - Keine Daten vom Socket erhalten, aber Timeout -> Proxy fordert noch mehr Daten von der Browser Seite an

Schutzpotenzial

- Nutzung von IMAP_SSL und SMTP_SSL (Port 587 statt 25)
- Maildir beim Beenden des Programms löschen
- Anzahl der E-Mails verringern

Vielen Dank
