

# Špecifikácia projektu HTTP File Sniffer

Peter Pokojný

4. novembra 2010

## Abstrakt

HTTP File Sniffer (HFS) je desktopová aplikácia, ktorá používateľovi umožní odchytať ním špecifikované dáta prenášané cez HTTP protokol.

## 1 Princíp fungovania

1. Používateľ zadá filtre na žiadané dáta. Tieto budú obsahovať najmä masku hostname-u serverov, masku MIME typov.
2. Používateľ spustí proces odchyťovania dát.
3. Aplikácia pozoruje prenášané packety, z nich vyfiltruje HTTP komunikáciu, rekonštruuje jednotlivé TCP streamy, dekoduje na nich HTTP protokol. Dátový obsah každej HTTP komunikácie, ktorá splňa používateľom zadané filtre, bude ukladaný na disk.
4. Používateľ ukončí proces odchyťovania dát.

## 2 Technológia

### 2.1 Získavanie dát

HFS bude na prístup k prenášaným packetom používať knižnicu winpcap prostredníctvom jej .NET rozhrania sharpccap. Knižnica winpcap umožňuje základné filtrovanie odchyťovaných ethernetových rámcov, napríklad typ packetu (pre potreby HFS vždy IPv4 alebo IPv6), typ vyššej dátovej vrstvy (pre potreby HFS vždy TCP), TCP port (pre HFS väčšinou 80 - defaultný port pre HTTP - ale používateľ bude môcť zdefinovať vlastné porty, na ktorých bude chcieť sledovať traffic). Do aplikácie teda prídu čisté dáta ethernetových rámcov, ktoré reprezentujú TCP traffic nad IPv4 alebo IPv6 na špecifikovaných portoch.

### 2.2 Rekonštrukcia TCP streamov

Aplikácia bude musieť z ethernetových rámcov reprezentujúcich TCP packety rekonštruovať TCP streamy. Na túto funkčnosť existuje .NET knižnica, autor ktorej ju iba prepísal z JAVA (z opensource projektu Wireshark) do .NET. Táto knižnica ale nerozlišuje smery komunikácie a oba streamy TCP spojenia (in a out) zlučuje do jedného streamu. HFS by mal modifikáciou alebo úplným prerobením tejto knižnice TCP packety rekonštruovať priamo do dvoch streamov pre každé TCP spojenie.

### **2.3 Rozpoznávanie HTTP protokolu**

HFS v nadobudnutých TCP streamoch následne vyhľadá HTTP spojenia a bude ich procesovať, teda musí rozpoznávať jednotlivé HTTP Requesty a k nim nájsť príslušné HTTP Response. Tieto dáta prejdú do ďalšej úrovne, kde sa bude diať samotná filtrácia a ukladanie prenášaných dát.

### **2.4 Filtrovanie HTTP Requestov a Response-ov**

Táto vrstva dostane už naparse-ované HTTP Requesty a Response-y. Ak tieto spĺňajú používateľom zadané filtre, HFS sa z nich pokúsi vyextrahovať: MIME typ dát, názov súboru a samotné dáta, ktoré uloží na disk.

### **2.5 Obmedzenia**

HFS prirodzene nebude schopný dekódovať šifrovanú komunikáciu (HTTP cez SSL alebo TLS). Taktiež bude existovať limitácia v HTTP použitých transfer-encodingov. Základné transfer encodingy (ako plain, chunked, base64) budú podporované. V ideálnom prípade sa doimplementuje podpora aj na gzip a ďalšie najčastejšie používané encodingy.

## **3 Vývojové a aplikačné prostredie**

HFS bude vytvorený ako .NET 4.0 aplikácia, ako vývojové prostredie bude použité Visual Studio 2010 Professional Edition. Aplikácia pobeží na systémoch, kde je nainštalovaný .NET 4.0 a knižnica winpcap.