Turris

Opensource router

Michal Hrušecký • Michal.Hrusecky@nic.cz

Kdo, co a proč?

CZ.NIC

- správce české domény
- vývoj pro blaho internetu
  - Knot - DNS server a resolver
  - BIRD - internet routing daemon (BGP, OSPF, …)
  - …
- osvěta v oblasti IT
  - knihy a televizní pořady
  - kurzy
- Český národní CSIRT tým

Jak to začalo

Otázka:

Jak nebezpečné je pro běžné lidi být připojen k interetu?

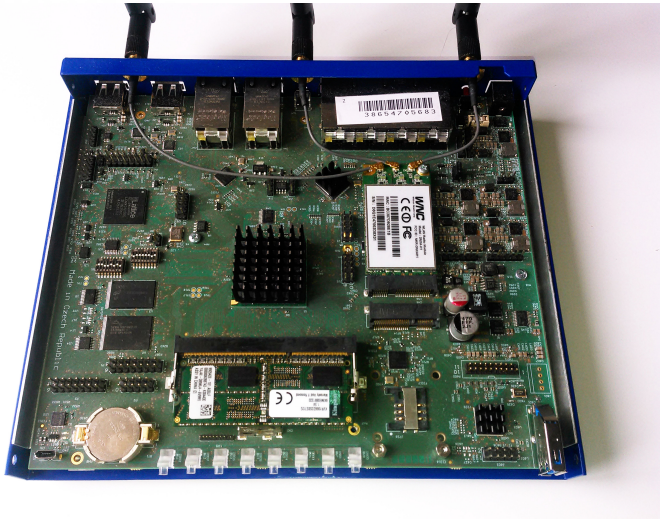
Útočí někdo i na domácnosti?

Jak často? Jednou za čas omylem? Pořád a cíleně?

Co za útoky můžeme pozorovat u běžného Franty uživatele?

První Turris router

- Cíl - experiment s bezpečným routerem
  - rozdán v ČR
  - automatické aktualizace
  - koncové body honeypotu
  - sběr dat o útocích
  - root account
- použita distribuce OpenWrt
  - systém nám spravuje komunita
  - složitější funkce v LuCI
  - soustředili sme se na naše funkce



Dnešní Turris routery

- cílový zákazník chce bezpečný router co umí víc
- vlastní uživatelské rozhraní
  - víc user-friendly než LuCI, jiný koncept
- integrace různých third-party aplikací
  - NetData, Nextcloud, Transmission, …
- na bezpečnostních aktualizacích spolupracujeme s komunitou

Výhody opensource pro nás

- snadný start projektu
- mnoho dostupného software
- mnoho funkcí lze dát dohromady vcelku snadno
- komunita přispívá a opravuje své problémy
- komunita přispívá i nové funkce
- bezpečnost

Bezpečnost

In-house security bude mít vždy méně CVE

Neznamená to, že je bezpečnější

- míň očí míň vidí
- 100% chyb bude objeveno ve vašem produktu

Opensource:

- lidi dělají bezpečnostní audit aniž byste jim museli platit
- chyby se nacházejí a opravují "samy"
- typicky se chyba najde nejdřív u někoho jiného

Příklad: Pakoň

- monitorovací nástroj pro sledování provozu na síti
- založen na Suricata IDS
  - heavy-lifting dělá opensource backend
- naše přidaná hodnota - ukládání a prezentace dat
- nové verze Suricaty závisí na Rust
  - aktuálně nemáme a stálo by to hodně úsilí
  - máme několik možností co dál
    - přejít na jiný backend - existují jiné projekty
    - upravit Suricatu aby fungovala bez Rustu
    - naportovat si Rust

Nevýhody opensource pro nás

- mnoho dostupného software
  - zákaznící chtějí ještě víc
  - předpokládají že všechno známe a umíme opravit
- externí přerušení
  - nové verze vycházejí bez ohledu na naše plánování
  - CVE může přijít kdykoliv a je třeba ho zpracovat
- lidi se v SW více hrabou

Upstreaming

Způsob jak se zbavit údržby upravených částí projektu - poslat patche upstream projektu.

- musí být správná doba v rámci release cyklu
- musí být dostatečně obecné
- musí splňovat upstream normy
- budou si dál žít vlastním životem
- zabere to čas
- opensource komunita to ráda vidí
  - ideálně se to mělo stát včera

Závěr

Výhody:

- snadnější začít
- snadněji se přidávají nové funkce
- dost často dostupné alternativy
- bezpečnost

Nevýhody:

- vývoj se dá ovlivnit pouze aktivním zapojením
  - neřídí se našimi plány
  - může i zaniknout nebo nabrat "špatný" směr