



AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE  
AGH UNIVERSITY OF KRAKOW

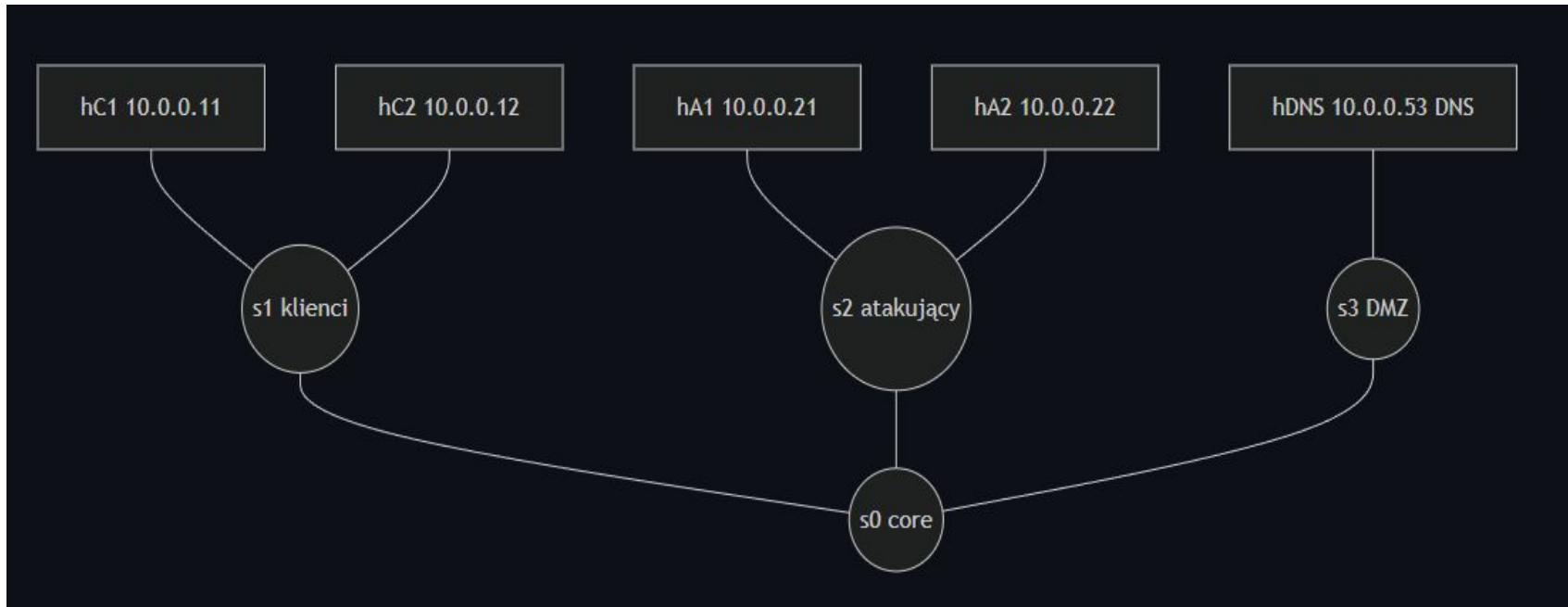
# Reaktywny Firewall w Floodlight przy atakach DNS Flooding

Piotr Gwardys  
Andrzej Filipowski  
Wiktor Kowalski  
Tymoteusz Pilarz

# Cel projektu

- Wykrywanie ataku DNS flooding
- Blokada adresów IP atakujących hostów

# Topologia



# Narzędzia

- **dig**
- **dnsmasq**
- **bash**
- **mininet**
- **floodlight na VMce**

# Algorytm

**Algorithm 1** Hybrydowy model wykrywania DNS Flooding (per-domena + entropia)

---

**Require:** Strumień zapytań DNS, okno czasowe  $\Delta t$ , liczba okien referencyjnych  $W$ , współczynniki  $k_N, k_H$

**Ensure:** Decyzyja o wystąpieniu ataku DNS Flooding

- 1:  $L_N[d] \leftarrow \emptyset$  {Historia liczby zapytań dla domeny  $d$ }
- 2:  $L_H[d] \leftarrow \emptyset$  {Historia entropii dla domeny  $d$ }
- 3: **while** system aktywny **do**
- 4:    $Q \leftarrow$  zbiór zapytań DNS w oknie  $\Delta t$
- 5:   **for all** domena bazowa  $d$  w  $Q$  **do**
- 6:      $S_d \leftarrow$  zbiór subdomen domeny  $d$  w  $Q$
- 7:      $N_d \leftarrow |S_d|$
- 8:      $H_d \leftarrow -\sum_{i=1}^{|S_d|} p_i \log_2 p_i$  {Entropia Shannona subdomen:  $p_i$  to prawdopodobieństwo subdomeny  $i$ }
- 9:     dodaj  $N_d$  do  $L_N[d]$
- 10:    dodaj  $H_d$  do  $L_H[d]$
- 11:    **if**  $|L_N[d]| \geq W$  **then**
- 12:      $\mu_N \leftarrow \frac{1}{W} \sum_{i=1}^W L_N[d]_i$  {Średnia liczby zapytań}
- 13:      $\sigma_N \leftarrow \sqrt{\frac{1}{W} \sum_{i=1}^W (L_N[d]_i - \mu_N)^2}$  {Odchylenie standardowe}
- 14:      $\mu_H \leftarrow \frac{1}{W} \sum_{i=1}^W L_H[d]_i$  {Średnia entropii}
- 15:      $\sigma_H \leftarrow \sqrt{\frac{1}{W} \sum_{i=1}^W (L_H[d]_i - \mu_H)^2}$  {Odchylenie standardowe}
- 16:      $TH_N \leftarrow \mu_N + k_N \cdot \sigma_N$  {Próg dla liczby zapytań}
- 17:      $TH_H \leftarrow \mu_H + k_H \cdot \sigma_H$  {Próg dla entropii}
- 18:     **if**  $N_d > TH_N \vee H_d > TH_H$  **then**
- 19:       zgłoś atak DNS Flooding dla domeny  $d$
- 20:     **end if**
- 21:     usuń najstarszy element z  $L_N[d]$
- 22:     usuń najstarszy element z  $L_H[d]$
- 23:   **end if**
- 24:   **end for**
- 25: **end while**

---

# DEMO

prosze działać...