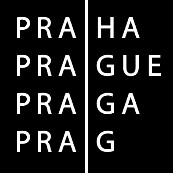
**Střední průmyslová škola elektrotechnická a gymnázium V Úžlabině 320**



**Maturitní práce z odborných předmětů**

**Název práce v českém jazyce**

**Jméno a příjmení autora/autorky práce Vyberte vaši třídu**

Vedoucí práce: **Ing. David Čepička** Studijní obor: **Informační technologie**

Rok odevzdání: **2025**

**Čestné prohlášení**

Odevzdáním této maturitní práce na téma **Útoky typu DoS** potvrzuji, že jsem ji vypracoval/a pod vedením vedoucího samostatně za použití v práci uvedených pramenů a literatury. Dále potvrzuji, že odevzdaná vytištěná verze písemné zprávy (protokolu) a plakátu se plně shoduje s odevzdanou elektronickou verzí.

V Praze dne ..................................... ...................................................

Podpis autora/autorky práce

**Anotace**

**Klíčová slova**

Kyberbezpečnost, DoS, HTTP, TCP,

**Keywords**

Cybersecurity, DoS, HTTP, TCP,

**Obsah**

[1 Úvod a cíl práce 4](#_Toc190766486)

[1.1 Cíl a zaměření práce (smazat?) 4](#_Toc190766487)

[1.2 Význam problematiky útoku typu „Denial of Service“ (smazat?) 4](#_Toc190766488)

[1.3 Struktura dokumentu (smazat?) 4](#_Toc190766489)

[2 Název kapitoly – teoretická část práce 4](#_Toc190766490)

[2.1 Princip a funkce útoků typu DoS 4](#_Toc190766491)

[2.2 Základní členění DoS útoků 5](#_Toc190766492)

[2.2.1 Distribuované útoky 5](#_Toc190766493)

[2.2.2 Volumetrické útoky 6](#_Toc190766494)

[2.2.3 Protokolové útoky 6](#_Toc190766495)

[2.2.4 Aplikační útoky 7](#_Toc190766496)

[2.3 Známé nástroje pro realizaci útoků 8](#_Toc190766497)

[2.3.1 LOIC 8](#_Toc190766498)

[2.3.2 Hping3 9](#_Toc190766499)

[2.3.3 HULK 9](#_Toc190766500)

[2.4 Prevence a obrana 9](#_Toc190766501)

[2.4.1 Firewally a IPD/IDS systémy 10](#_Toc190766502)

[2.4.2 Load balancery 10](#_Toc190766503)

[2.4.3 Protokoly 10](#_Toc190766504)

[2.4.4 Prevence ve firmách 11](#_Toc190766505)

[3 Praktická část? – vlastní práce 11](#_Toc190766506)

[3.1 Popis virtualizovaného prostředí 11](#_Toc190766507)

[3.2 Konfigurace prvků služeb 12](#_Toc190766508)

[3.3 Set-up scénůřů 13](#_Toc190766509)

[3.4 Scénář – HTTP flood 14](#_Toc190766510)

[3.5 Scénář – TCP SYN flood 14](#_Toc190766511)

[4 Závěr 15](#_Toc190766512)

# Úvod a cíl práce

## Cíl a zaměření práce (smazat?)

## Význam problematiky útoku typu „Denial of Service“ (smazat?)

## Struktura dokumentu (smazat?)

Struktura dokumentu se dělí na úvod práce, teoretickou a praktickou část práce a její závěr. Teoretická část práce obsahuje základní řazení DoS útoků, nástroje k jejich nasazení a následnou prevenci. Praktická část se naopak zaobírá realizací dvou scénářů, jejich topologii a útoky, které byly použity.

# Název kapitoly – teoretická část práce

Než začneme s typy DoS útoků a možnou obranou proti nim, je třeba nejprve pochopit principy těchto útoků a jejich fungování. Až dále si ukážeme jaké nástroje se používají pro jejich realizaci a jaké jsou možné varianty obrany proti nim.

## Princip a funkce útoků typu DoS

Útoky typu DoS (Denial of Service) jsou zaměřeny na znepřístupnění online služeb prostřednictvím přetížení cílového systému nadměrným množstvím síťového provozu. Tento nápor vede k vyčerpání zdrojů, což způsobí, že systém pak není schopen obsloužit legitimní požadavky uživatelů.

Většinou se u DoS útoků setkáváme se zaplavováním (flooding), což je odesílání velkého množství dat nebo požadavků s cílem přetížit šířku pásma (bandwidth) nebo výpočetní kapacitu cílového zařízení. Anebo také s využitím nějaké slabiny (exploit) v síťových protokolech nebo aplikacích.

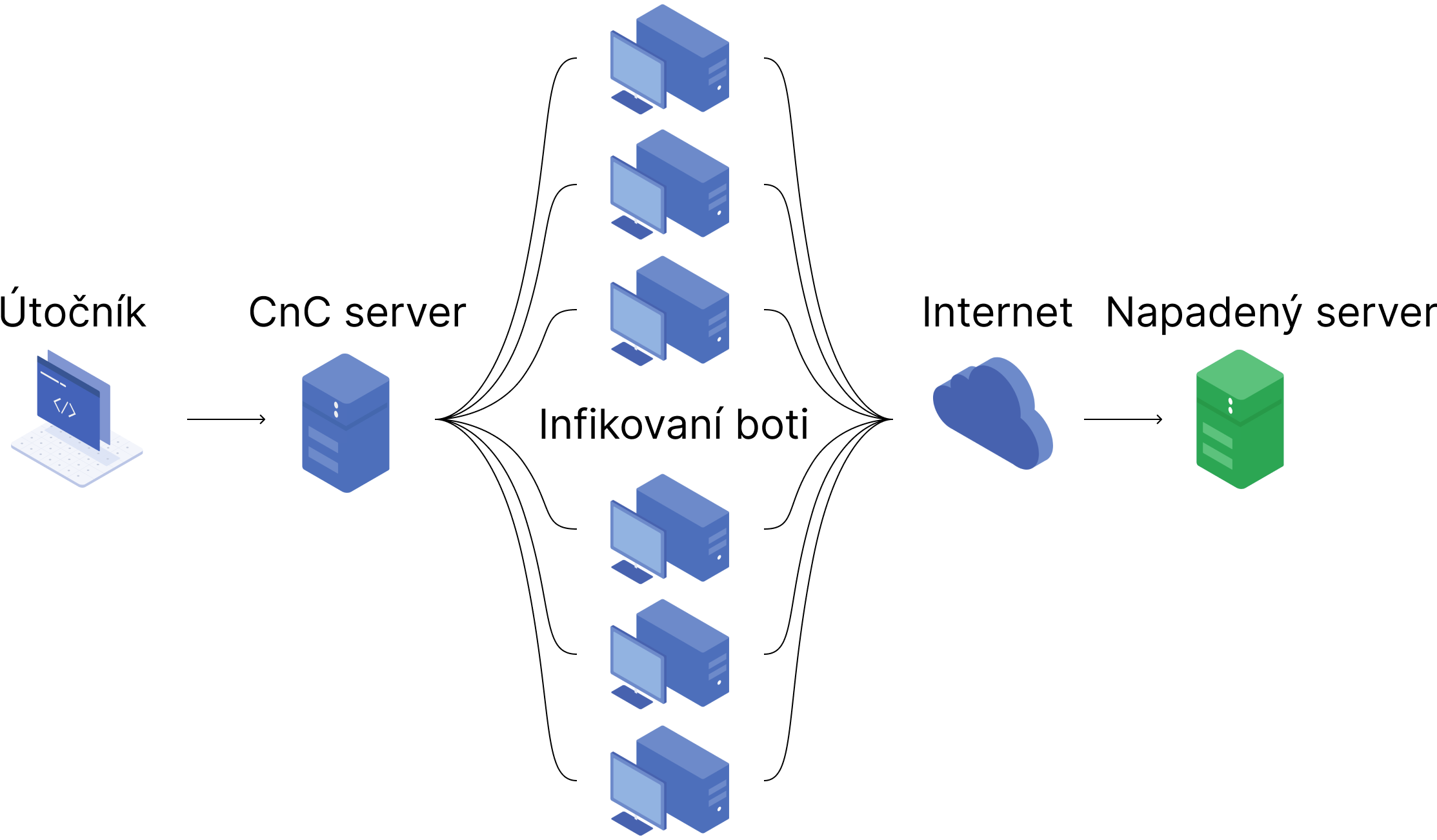
Důsledkem útoků DoS jsou finanční ztráty, narušení reputace anebo ztráta důvěry zákazníků. S rostoucím počtem těchto útoků je stále naléhavější potřeba implementovat robustnější bezpečnostní strategie. (example růstů <https://www.researchgate.net/publication/378952021_Evolving_Malware_DDoS_Attacks_Decadal_Longitudinal_Study>, nebo <https://nukib.gov.cz/download/publications_en/2023_Report_on_the_State_of_Cybersecurity_in_the_Czech_Republic.pdf>)

## Základní členění DoS útoků

### Distribuované útoky

Distribuované útoky typu Denial of service, se liší od normálních DoS útoků tím, že odcházejí z více lokací najednou, a tak účinnost útoků je vyšší. DDoS útoky svým velikým počtem dat maskují útočníka a také rychlost jakou mohou cíl vyřadit z provozu je mnohonásobně větší.

Pro lepší zdůraznění si můžeme ukázat základní popis sítě DDoS útoku na figuře 1.



Obrázek Schéma základního DDoS útoku – vlastní zpracování

Na obrázku 1 vidíme použití strategie botnetu, což je kolekce botů (infikovaných počítačů), kteří jsou většinou pod hierarchickou kontrolou. Na konci této kontroly se většinou vyskytuje také Command-and-Control (CnC) Server, který rozesílá útočníkovi instrukce svým botům.

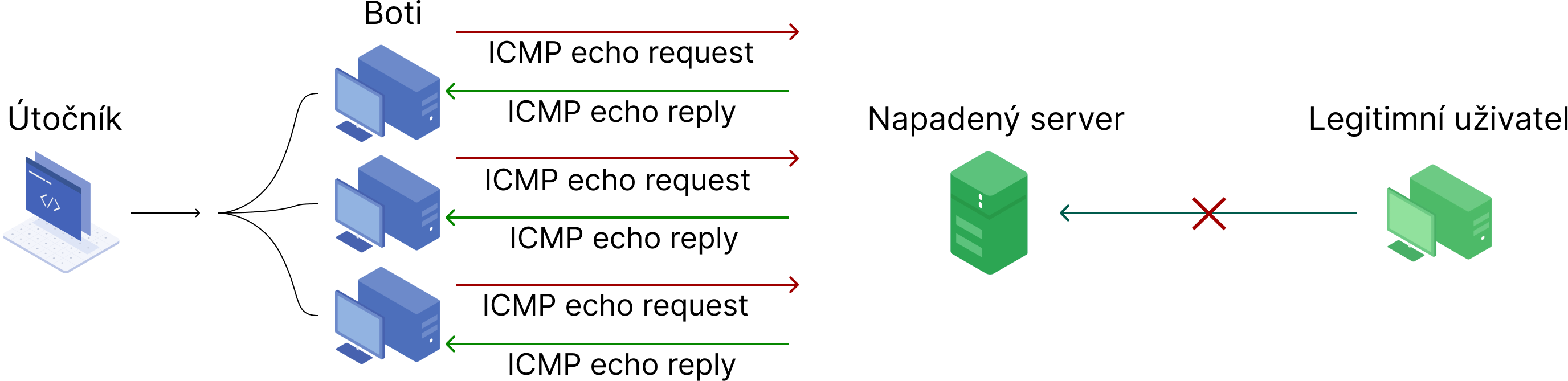
Dokazují to i tyto informce…/Další informace k tomuto tématu lze nalézt v dokumentu Lecture 29, Purdue University. ? <https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture29.pdf>

### Volumetrické útoky

„Volumterické útoky jsou nejčastějším typem DDoS útoků. Jejich cílem je přesycení šířky pásma cíle, což způsobí nedostupnost služby. Tyto útoky zahlcují cíl obrovským množstvím dat, které se často měří v bitech za sekundu (Bps) nebo gigabitech za sekundu (Gbps).“ (Merkebaiuly, 2024)

U volumetrických útoků je i složitější rozpoznat legitimní uživatele/požadavky od škodlivého provozu, protože velký objem dat zahlcuje nejen bezpečnostní prvky v síti, ale i prvky na napadeném zařízení, které by je mohli rozlišit.

Volumetrické útoky se pohybují primárně ve třetí a čtvrté vrstvě ISO/OSI modelu. Do těchto útoků spadá například UDP flood, SYN flood, ICMP flood nebo DNS reflection flood útoky. (Merkebaiuly, 2024)



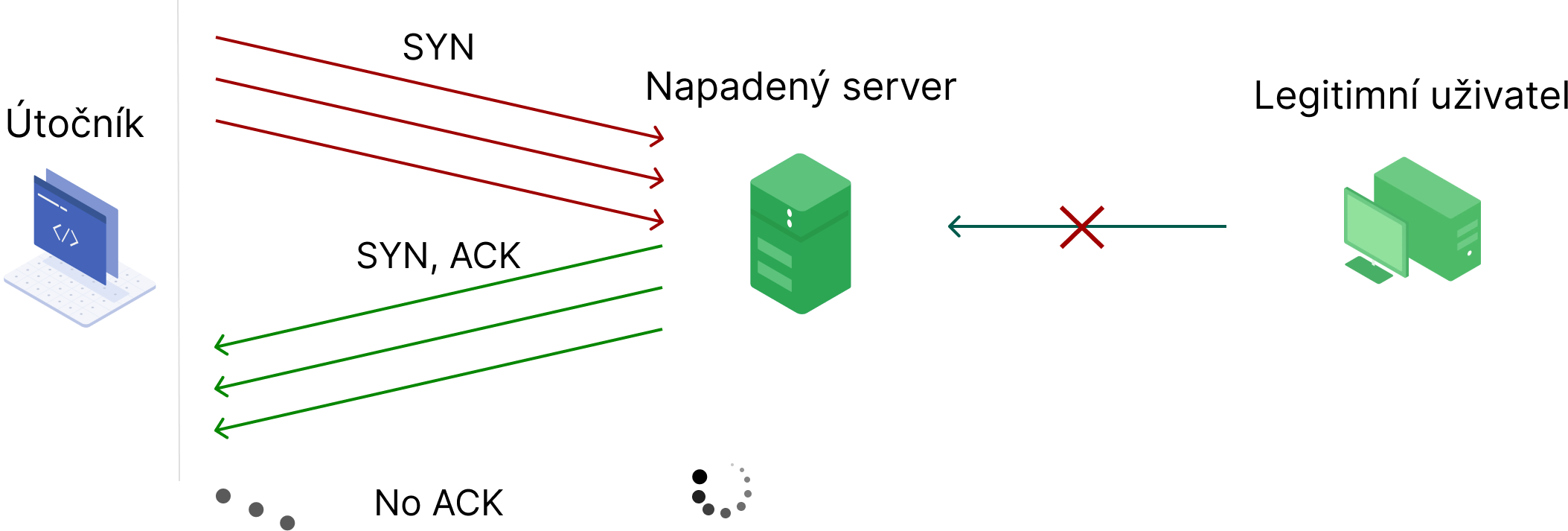
Obrázek Schéma ICMP flood útoku – vlastní zpracování

Na levé straně figury 2, která znázorňuje ICMP flood útok, můžeme vidět útočníka řídící botnet a samotnou záplavu ICMP echo požadavky. Napadený server odpovídá a útok tak vyplýtvá jeho šířku pásma (bandwidth). Na pravé straně je zase legitimní uživatel, který není schopen komunikovat s přetíženým serverem.

### Protokolové útoky

Protokolové útoky se také pohybují ve třetí a čtvrté vrstvě ISO/OSI modelu, ale na rozdíl od volumetrických se zaměřují na využití slabin v protokolu, například v handshake procesu nebo navázání relace. (Merkebaiuly, 2024)

Nejčastěji se tyto útoky měří v (Pbs) paktetech za sekundu. Do těchto útoků pak spadá TCP SYN flood, Smurf útoky nebo útok ping of death.

**

Obrázek Schéma TCP SYN flood útoku – vlastní zpracování

TCP SYN flood funguje na základě operace „three-way handshake“, která se používá k navázání TCP komunikace. Normální průběh této operace probíhá tak, že klient pošle synchronizační zprávu (SYN), server odpoví také synchronizační zprávou (SYN) a ještě uznáním klientovi zprávy (ACK) a nakonec klient nazpět uzná serveru jeho zprávu (ACK).

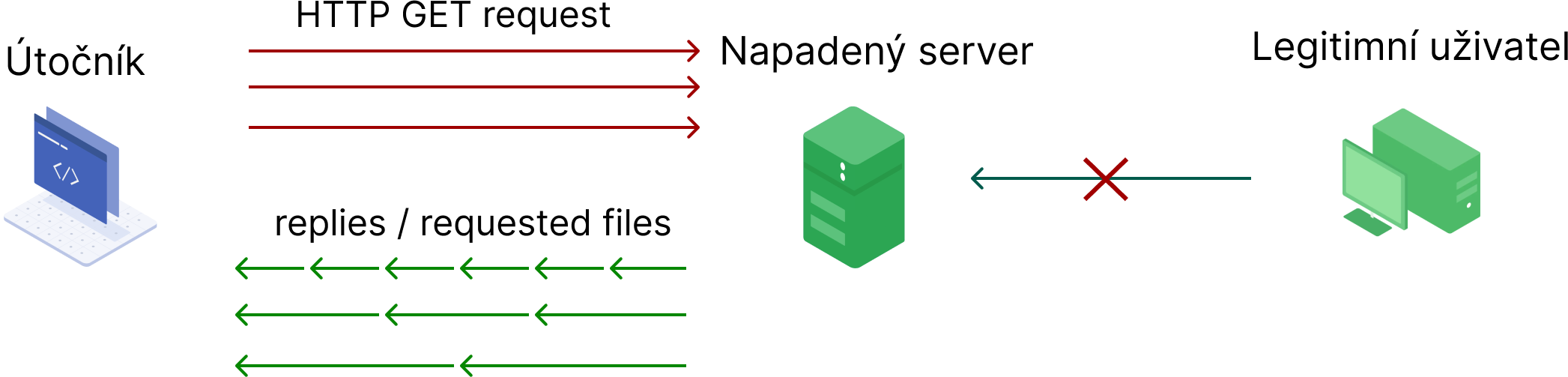
Trik ve TCP SYN flood útoku je ten, že pokud klient neodpoví poslední zprávou (ACK), tak instance navázání komunikace bude stále otevřená. Nakonec server může dojít až do kritického bodu (viz. Obrázek 3) kdy bude přetížen a nebude schopen odpovídat na další TCP SYN zprávy.

### Aplikační útoky

„Aplikační útoky se zaměřují na nejvyšší vrstvu ISO/OSI modelu. Jejich cílem je vyčerpat zdroje konkrétní aplikace. Tyto útoky je často obtížnější odhalit, protože napodobují legitimní požadavky a provoz.“ (Merkebaiuly, 2024)

Nicméně, tyto útoky budou mít podobný dopad na služby, protože cílí na specifické charakteristiky aplikací, jako jsou HTTP, DNS nebo protokoly pro inicializaci relací (SIP). (Zargar, 2013)

Mezi tyto útoky patří HTTP GET/POST flood útoky, BGP hijacking nebo Slowloris útoky.



Obrázek Schéma HTTP GET flood útoku – vlastní zpracování

Na figuře 4 je znázorněn právě HTTP GET flood útok, kde útočník posílá velké množství GET požadavků (requests), například na stáhnutí obrázků či scriptů. Tím zapříčiní přetížení serveru. Dále k celkovému útoku i napomáhá fakt, že server musí odpovídat většinou i objemnými daty. Útočníci také kombinují různé parametry nebo i celkovou strukturu útoku pro co nevyšší poškození.

## Známé nástroje pro realizaci útoků

### LOIC

Jeden z nejznámějších nástrojů k provádění DoS a DDoS útoků. Původně byl vyvinut společností Praetox Technology jako aplikace pro testování zatížení sítí, ale později se stal open-source softwarem a nyní se používá i pro škodlivé účely. LOIC je známý svou jednoduchostí a dostupností, což ho činí oblíbeným i mezi lidmi s minimálními technickými znalostmi. Proslavil se použitím skupinou Anonymous a uživateli online fór.

Nástroj umožňuje zaplavovat cílové servery s TCP, UDP nebo http pakety. Pro rozsáhlé útoky vyžadující spolupráci tisíce uživatelů je k dispozici mód „Hivemind“, kdy jeden hlavní uživatel řídí více připojených zařízení (dobrovalný botnet) prostřednictvím IRC chatovacích kanálů. Účastníci tak můžou tvrdit, že jejich zařízení byla oběťmi botnetu.

Mezi významné incidenty s využitím LOIC patří útoky Anonymous na weby Church of Scientology v roce 2008 a útoky podporovatelů WikiLeaks na weby Visa a MasterCard v roce 2010 poté, co tyto společnosti zablokovaly platby směrem k WikiLeaks. (Cloudflare, 2024)

### Hping3

Hping3 je síťový CLI nástroj, který dokáže odesílat vlastní ICMP/UDP/TCP pakety a zobrazovat odpovědi cíle podobně jako ping zobrazuje ICMP odpovědi. Podporuje fragmentaci, libovolné tělo a velikost paketů a lze jej použít k přenosu souborů prostřednictvím podporovaných protokolů.

Pomocí hping3 můžeme testovat pravidla firewallu, provádět (falšované) skenování portů, testovat výkon sítě s použitím různých protokolů, provádět objevování MTU cesty, provádět operace podobné traceroute s různými protokoly, identifikovat vzdálené operační systémy, nebo auditovat TCP/IP stacky. (Kali Hping3, 2024)

### HULK

HULK (HTTP Unbearable Load King) byl primárně navržen pro výzkumné účely a měl pomáhat penetračním testerům ověřovat efektivitu serverů. Důvod vzniku byl za vylepšení dřívějších penetračních nástrojů, jelikož většina generovala předvídatelné pakety nebo http SYN požadavky, což bránilo efektivně testovat obrany sítí.

HULK se tak liší od běžných nástrojů pro penetrační testování, skriptů nebo metod exploitace. Generuje množství unikátních požadavků v nepravidelných intervalech z jednoho hostitele. Nejenže spouští DDoS útok, ale zároveň ztěžuje obranným mechanismům sítě odhalit vzor útoku, což komplikuje filtrování provozu. Nástroj nabízí také funkce, jako je maskování referenčních požadavků a skrytí identity útočníka. (Beschokov, 2024)

## Prevence a obrana

Obvykle, když je útok DDoS typu flooding detekován pozdě, nelze udělat nic jiného než odpojit oběť od sítě a ručně problém vyřešit. Útoky DDoS typu flooding plýtvají značným množstvím zdrojů (například výpočetním časem atd.) na cestách vedoucích k cílovému zařízení. Proto je hlavním cílem jakéhokoli mechanismu obrany proti DDoS detekovat tyto útoky co nejdříve a zastavit je co nejblíže jejich zdrojům. (Zargar, 2013)

Avšak pro dokonalou obrobnu by se muselo synchronizovaně kontrolovat na každém kroku po internetu, kvůli rozsáhlosti DDoS útoků. A díky tomu, že implementace širokého nasazení obrany se nemůže vnutit na všechny uživatele internetu <https://www.princeton.edu/~rblee/ELE572Papers/Fall04Readings/DDoSmirkovic.pdf> str47

### Firewally a IPD/IDS systémy

Firewally a systémy pro detekci a prevenci průniků (IDS/IPS) představují základní vrstvu obrany proti DDoS útokům.

Firewally umožňují blokovat podezřelý síťový provoz na základě předem nastavených pravidel, jako je filtrování podle IP adres, portů nebo protokolů, a tak zabraňují před neautorizovaným přístupem.

IDS/IPS systémy přidávají schopnost detekovat a automaticky reagovat na škodlivý provoz. IDS systémy analyzují síťovou komunikaci a upozorňují na možné hrozby, zatímco IPS systémy dokáží aktivně blokovat škodlivé pakety v reálném čase. Tyto technologie jsou však limitovány svou schopností zvládat vysoké objemy provozu, které větší DDoS útoky generují. (Scarfone, 2007)

### Load balancery

Load balancery představují účinnou metodu pro distribuci síťového provozu mezi více servery. Tím se snižuje riziko přetížení jednoho zařízení a umožňuje lepší odolnost vůči DDoS útokům. V případě útoku může load balancer přesměrovat provoz na méně vytížené servery nebo na speciálně navržené servery pro absorpci DDoS útoků. Moderní load balancery mohou navíc využívat metody analýzy provozu k identifikaci a filtrování škodlivého provozu. (f5, 2023)

### Protokoly

Nastavení, optimalizace a analýza síťových protokolů také hrají klíčovou roli v obraně proti DDoS útokům. Například protokol TCP může být nakonfigurován tak, aby omezoval počet současných spojení nebo minimalizoval dobu čekání na odpověď. Podobně lze využít metody jako „rate limiting“, které kontrolují počet požadavků z jedné IP adresy v určitém čase.

Také jak píše Mirkovic et al. (2004), pokud se filtruje pomocí mechanismu, který ověřuje například TCP připojení, tak podle jeho definice může odhalit polootevřené spojení odpojit je a tím zabránit jejich hromadění.

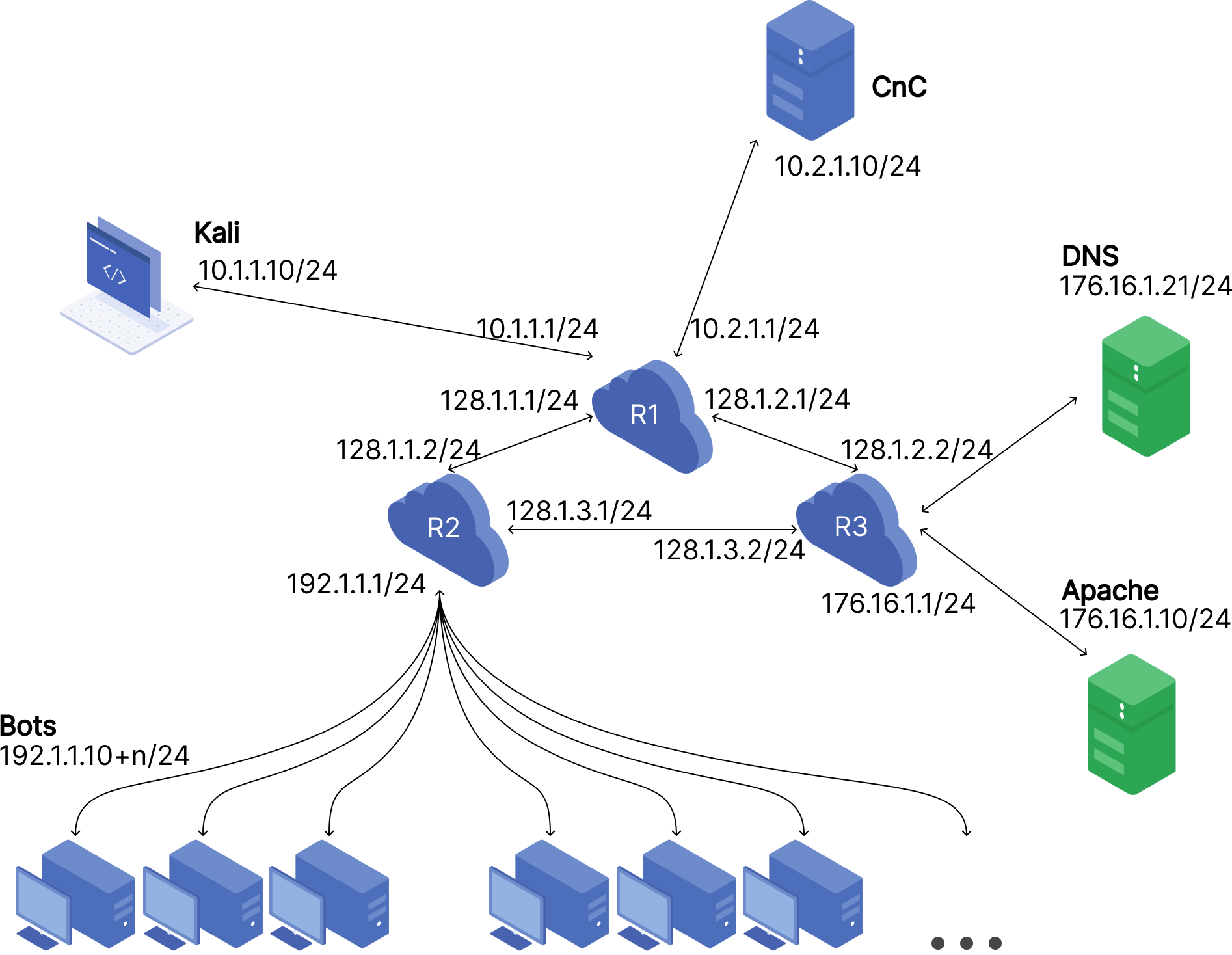
### Prevence ve firmách

Prevence DoS a DDoS útoků ve firmách zahrnuje kombinaci technologických opatření, procesů a osvěty zaměstnanců. Bezpečnostní opatření na síťové úrovni jsou například segmentování sítě či „geo-locking“, ale i ochranné systémy jako Next-Generation Firewally a už zmínění IPS. Dále se využívají i Webové Firewally (WAF) nebo API brány na ochranu API koncových zařízení před zneužitím nebo nadměrným zatěžováním. Určitě ale hlavní částí bezpečnosti ve firmách je i pravidelné školení zaměstnanců, plánování postupů na určité incidenty nebo pravidelné testování penetračními testy.

# Praktická část? – vlastní práce

Celá kapitola 3 je věnovaná dvou útokům, HTTP flood a TCP SYN flood, a celkovému rozložení prostředí, kde se tyto útoky budou odehrávat.

## Popis virtualizovaného prostředí



Obrázek 5 Topologie praktické části – vlastní zpracování

Na Obrázku 5 je celá topologie praktické části i s přiřazenými IP adresami. V tomto zapojení tedy bude komponovat Útočník, na kterém běží distribuce Kali (dostupná z: <https://www.kali.org/get-kali/#kali-virtual-machines>), ze kterého budou pramenit útoky. Centralizovaný CnC (Command and Control) server, ke kterému Útočník má root přístup a ze kterého se budou přeposílat útoky botnetu. Boti, kteří reprezentují infikované zařízení s přístupem na internet a jsou tedy spouštěči škodlivého kódu. Tři routery s operačními systémy RouterOS a Ubuntu (dostupný z: <https://mikrotik.com/download>), mají simulovat internet a umožnit load balancing, pokud je potřeba. A samotné servery oběti, na kterých běží HTTP služba Apache a DNS služba Bind na překlad adresy z druhého serveru. Jak Boti, tak servery běží převážně také na ubuntu (dostupný z: <https://ubuntu.com/download/server>).

## Konfigurace prvků služeb

Vše běží virtualizovaně v programu VirtualBox a komunikace je řešena přes interní sítě a komunikace do vzdálených sítí je řešena statickým routingem.

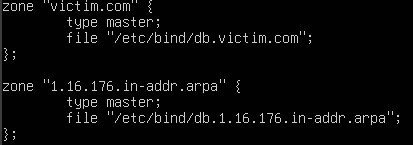
Router R1 s RouterOS má nastavený bridge pro rychlejší konfiguraci a setupování a je spravován pomocí programu WinBox. Na R1 je také nastavená src-nat NAT a DHCP client, což poskytuje komunikaci do internetu pro celou topologii.

Pro R2 a R3 se používá Ubuntu a pro konfiguraci jejich síťových parametrů se používá nástroj nmtui.

U všech koncových zařízení se rozhraní konfigurovali podle úpravy souboru ve „/etc/network/interfaces“, nebo za pomocí nástroje nmtui.

Na serveru, který poskytuje webovou službu stačí nastavit kořenovou složku ve „/etc/apache2/http.conf“ a upravit index.html abychom si byli jistí, že se změny vážně propsali. Nakonec službu můžeme spustit pomocí „sudo /etc/init.d/apache2 start“, nebo „sudo systemctl start apache2.service“, závisí jakou verzi operačního systému máme.

Konfigurace pro DNS server je tedy zprovozněna podle služby bind9, takže hlavní soubory k úpravě jsou ve složce „/etc/bind/“. Ve „named.conf.options“ stačí změnit na jaké IPv4 adrese budeme poslouchat a povolíme jakýkoliv provoz. V „named.conf.local“ (viz příloha 1) si už rozvrhneme pojmenování naší stránky (například victim.com). Pak v samotné „named.conf“ ověřit, že oba soubory jsou zahrnuty.



Příloha Obsah named.conf.local

Pak už stačí vytvořit a nastavit forward zónu „db.victim.com“ a reverse zónu „db.1.16.176.in-addr.arpa“. Pro rychlejší tvorbu se může využít i stránky jako yoyo.org na generování SOA a NS záznamu. Konečná konfigurace našich zón bude vypadat tedy takto – příloha 2 a 3.



Příloha Obsah db.victim.com



Příloha Obsah db.1.16.176.in-addr.arpa

Nakonec pokud je firewall aktivní tak službu dns přidáme s „firewall-cmd --permanent --add-service=dns“ a nakonec službu zapneme s „sudo /etc/init.d/named start“, nebo „sudo systemctl start named“.

## Příprava scénůřů

### Útočník - CnC

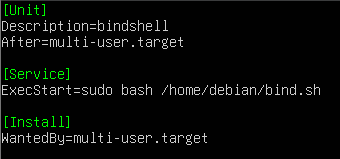
Ze zařízení útočníka se budeme připojovat na CnC server pomocí bind shell, za cílem stáhnout potřebný payload a spustit script, který řekne všem zaznamenaným botům, že si mají stáhnout payload a spustit ho.

Pro tento účel nám postačí one-liner bind shell script (například z PayloadsAllTheThings) nastavit jako „ .service” na CnC serveru, aby se spouštěl vždy při startu zařízení.



Příloha Bind shell script

Vytvořit a rovnou editovat service můžeme za pomocí „sudo systemctl --force --full edit <jméno>.service“.



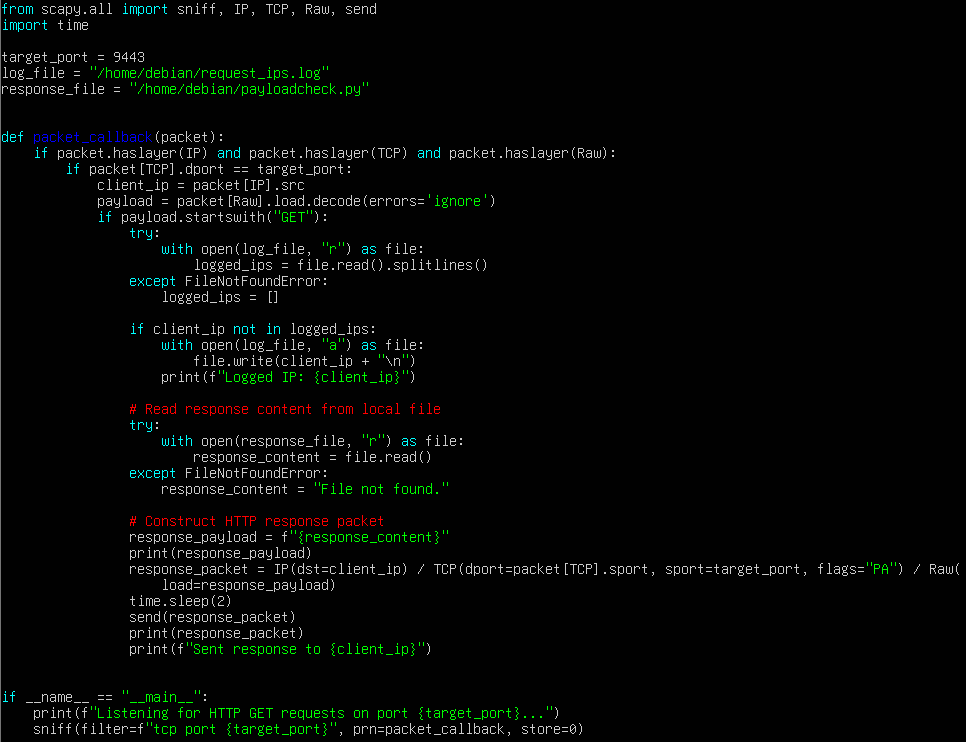
Příloha Obsah bind.service

Hlavní parametry v bind.service tedy jsou: doba spuštění a umístění souboru pro spuštění. Čas nám tady udává WantedBy=multi-user.target, který se spustí na runlevelu 2,3,4 (tedy při spuštění zařízení). A do ExecStart dáme celý command s absolutní cestou ke scriptu.

Na závěr musíme povolit (enable) náš vytvořený service, a pomocí status se ujistit, že je spuštěn. Zpětné připojení na bind shell jde pomocí nc příkazu na IP adresu CnC serveru a portu na kterém poslouchá.

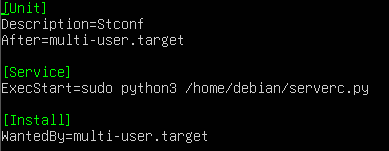
### CnC – Botnet

Dlaší script (viz Příloha 6), který chceme, aby běžel neustále od začátku spuštění je ten, který nám bude logovat IP adresy botnetu a posílat payload zpět. Logování IP adres využijeme hlavně když útočník chce, aby všechny zaznamenaní boti začali útočit navzájem.



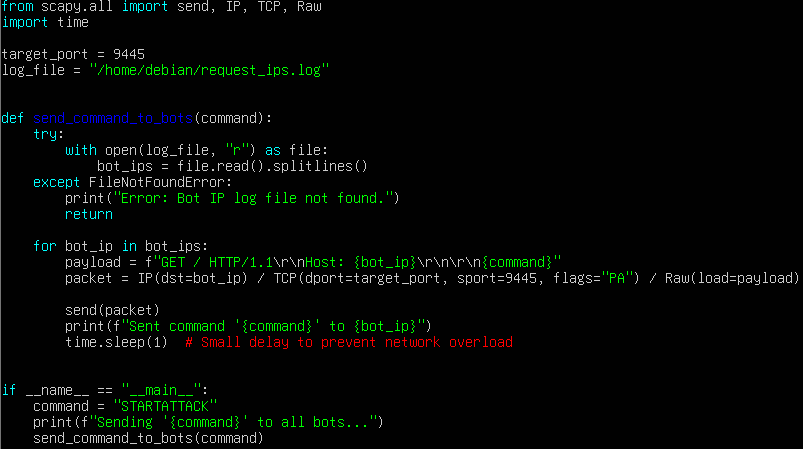
Příloha Hlavní script na CnC serveru

Pro posílání pakatů se využívá balíček scapy, kvůli dokumentaci a jeho nastavení. Celý script tedy porovná, jestli je packet celý a porovná cílový port s tím na kterém poslouchá. Log-ne jeho IP adresu a dále si uloží jeho obsah. Pokud je packet GET a zaznamená jeho IP adresu do .log souboru, pokud tam ještě není a načte obsah payload souboru. Dále na stejnou IP adresu a port pošle packet, ve kterém je obsažen náš škodlivý kód. Tento script běží neustále (také běží jako .service (viz Příloha 7)) a vlastně slouží jako „odchytáváč“ žádostí z botnetu.



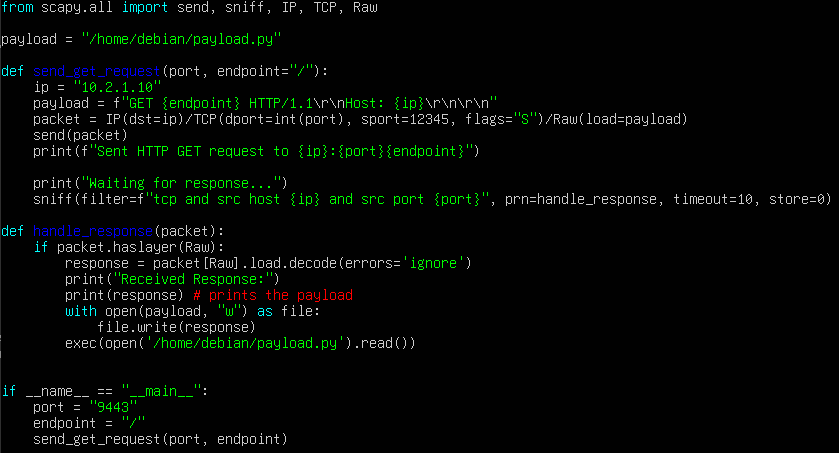
Příloha Service sniff scriptu na CnC serveru

Poslední script, který je uložen na CnC serveru je manuální „ping“ botnetu (viz Příloha 8) neboli našich zaznamenaných IP adres. Script projede každou adresu v našem .log souboru a pro každou vytvoří specifický packet s kódem, který pak pošle. Tím útočník může manuálně začít globální útok, který jde lehce škálovat.



Příloha Script na manuální spuštění útoku

### Botnet – Servery se službami

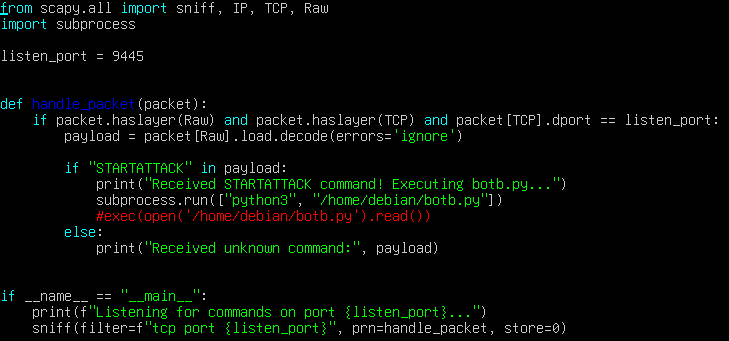


Příloha Script – žádost o payload

Základní script (viz Příloha 9), který běží na každém botu je tedy naše žádost o payload k CnC serveru. Spustí se jak při startu zařízení (využití .service), tak i pokud útočník použije script na vyžádání žádosti z Přílohy 8, což je postaráno skrze sniffer script (viz příloha 10).

Funguje tak, že pošle žádost na CnC server, u kterého už musí vědět IP adresu, který mu odpoví payload souborem. Hned po poslání žádosti začne čekat na odpověď a pokud ji dostane tak si ji uloží do payload.py a spustí. Tím prakticky může spustit jakýkoliv script.

Dále musíme mít script na odposlech „pingu“ botnetu od CnC serveru, který nám akorát dá koloběh do procesu spuštěním našeho hlavního scriptu. Aby se náhodou nestalo, že by se tento útok mohl provést z ničeho nic, tak jednoduchou podmínkou zjistíme jestli „ping“ paket obsahuje náš string a pokud ano, tak až teď spustí hlavní kód.



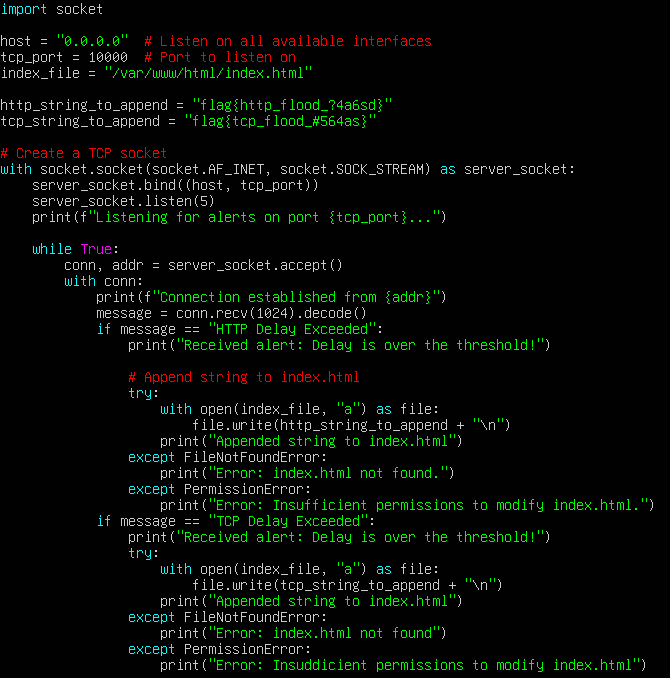
Příloha Sniffer script na spuštění žádosti

### Ověření funkčnosti

Ověření, zda útoky navyšují zpoždění můžeme zjisti buď přes ping na adresu serveru na či můžeme použít další kód na ověření, která nám rovnou na stránku může i přidat náš řetězec za splnění dostatečné podmínky/úrovně zpoždění.

Stačí nám tedy script, který bude odposlouchávat na portu, a pokud dostane zprávu, že zpoždění je vyšší než nastavená hodnota, tak přidá na konec index.html tajný řetěz (flag)(viz Příloha 11).

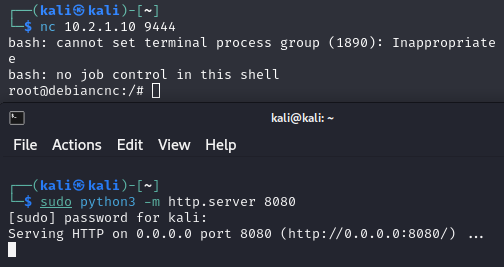
K němu na nevinném zařízení v topologii (pro ušetření místa můžeme klidně použít stroj útočníka) tedy poběží další script na kalkulaci a znázornění zpoždění a následné poslání zprávy, že překročila hranici.



Příloha Listen script na Apache serveru pro objevení řetězce

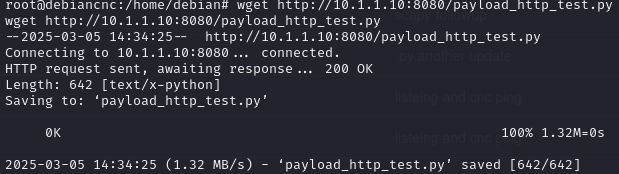
## Scénář – HTTP flood,

Scénář tedy začíná u útočníka (už tedy máme nakažené stroje, které se po spuštění zaznamenali do CnC serveru, například boti 1-4 s IP adresami 192.1.1.11-14), který se připojí k CnC serveru skrze bind shell. Zároveň si spustí http server u sebe, aby na CnC server mohl nahrát svůj libovolný script.



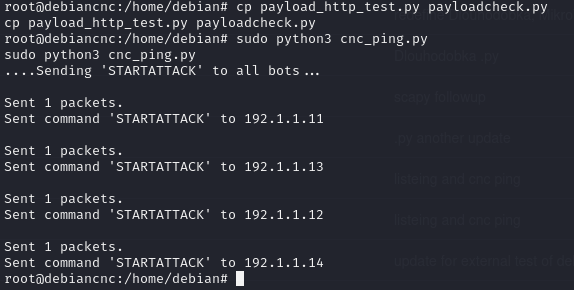
Příloha Bind shell + http server

Pak na CnC serveru si stáhneme náš payload pomocí nástroje wget.



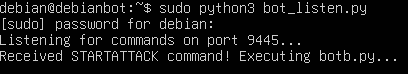
Příloha Wget http payload souboru

Ten přejmenujeme na soubor, který se bude šířit, což pro nás je teď payloadcheck.py a spustíme náš script, který upozorní všechny zaznamenané boty.



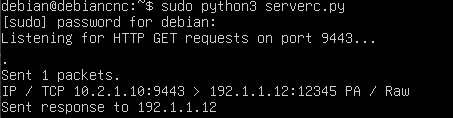
Příloha cnc\_ping-py

Tím se na všech uvedených zařízení aktivuje podmínka na jejich listen scriptu. Ten spustí další script, který si zažádá o payload ze CnC serveru. (Všechny další přílohy na CnC serveru a botnetu jsou spouštěny manuálně, za účelem ukázat chod útoku, jinak se spouštějí automaticky pomocí .service)



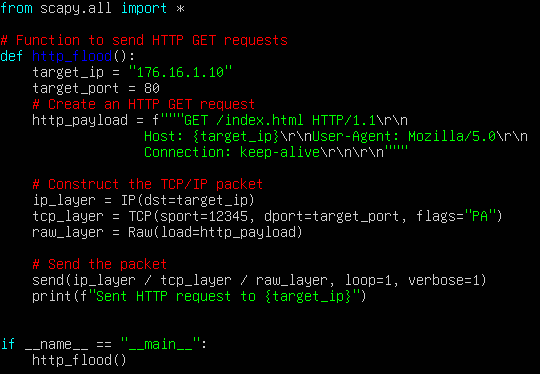
Příloha Přijetí zahájení útoku

Žádost se tedy dostane na CnC server, kde tedy se zaznamená IP adresa a ověří se paket. Pak se pošle HTTP zprávou payload zpět kontaktovanému zařízení. Na Příloze 16 je vidět, že CnC byl kontaktován botem s adresou 192.1.1.12 a jak odpovídá zpět.



Příloha Spuštění CnC scriptu

Jakmile bot obdrží paket, tak si jeho obsah uloží a uložený soubor spustí. V tomto scénáři použijeme tedy HTTP flood script (viz Příloha 17).



Příloha Script na HTTP flood útok

Znova využijeme balíček scapy, nadefinujeme si na jakou IP adresu budeme útočit, na jaký port, zkonstruujeme obsah paketu, který by měl vypadat obecně a pošleme sestavený paket přes funkci send(). Loop = 1 nám zajistí, že budeme posílat pakety, než se přístroj vypne, čí se proces ukončí.

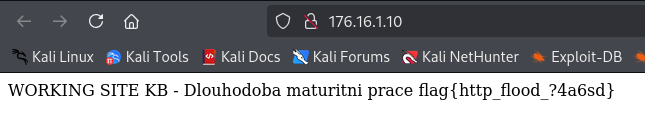
Na volitelném počítači (nejlépe Kali, kde se dá zobrazit stránka v prohlížeči) můžeme spustit script na měření odezvy (viz Příloha 18). Ten nám po změření vypíše výslednou hodnotu, dále se koukne, jestli je odezva větší než nastavená hranice. Pokud ano, tak na Apache server pošle, že ji útok úspěšně překonal.



Příloha Script na měření zpoždění

(Z mé zkušenosti stačí ~4 počítače v botnetu aby se odezva začala pohybovat namísto ~0.01 okolo 0.1–0.7 sekund. Více počítačů zapříčiní, že útok vyplýtvá výpočetní výkon (testováno na stroji s 32 GB RAM a 3.6 GHz i9 CPU), méně a server se z útoku dokáže docela dobře zotavit. Na usnadnění se může zpomalit server ve VirtualBoxu, kde se může přiřadit i kolik procent z CPU virtuální stroj může použít. Ale velkou změnu to v mém případě moc neudělalo.)

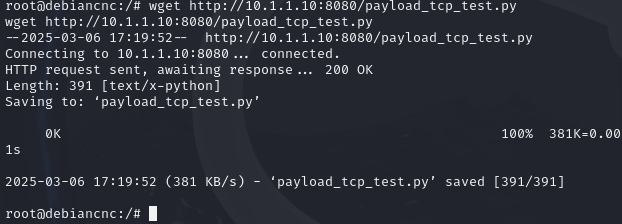
Následně po zaslání zprávy, kterou zpracuje script z Přílohy 11, se přidá na konec stránky příslušný řetězec.



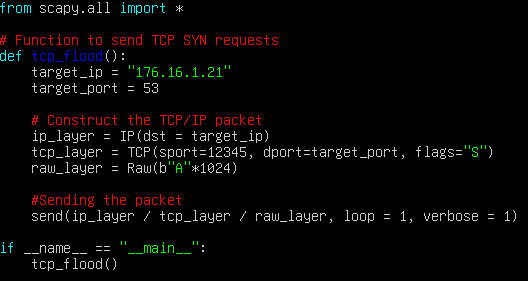
Příloha Splnění úlohy – HTTP řetězec

## Scénář – TCP SYN flood

Druhý scénář používá stejnou topologii i stejný princip útoku jako předchozí HTTP flood. Začátek tedy je stejný, akorát si stáhneme jiný script z Kali na CnC server. (postupujeme tedy stejně se založením lokálního serveru a spustíme CnC „ping“ script)



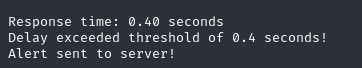
Příloha Wget tcp syn payload souboru



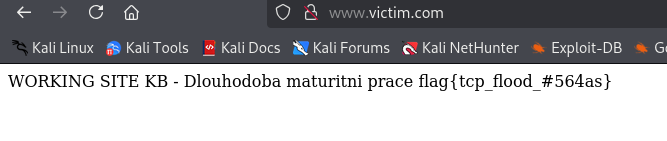
Příloha Script na TCP SYN flood

Obsah scriptu je také podobný jako u minulé úlohy. Nedeklarujeme si cílovou IP adresu a port, vytvoříme pomocí balíčku scapy TCP paket, který má naše deklarované hodnoty a neustále posíláme. To zapříčiní, že DNS server portu 53 bude přetížen a neměl by být schopen tak vykonávat jeho službu. Tím pádem normální uživatele, kteří na tuto stránku jdou poprvé (nevědí IP adresu stránky [www.victim.com](http://www.victim.com)), nebudou schopni se na stránku dostat.

Spustíme podobný script na ověření odezvy a po nějaké době by měla hodnota překročit požadovanou hodnotu. Tím se přidá na stránku náš řetězec pro splnění úlohy.



Příloha Script na ověření odezvy



Příloha Splnění úlohy – TCP řetězec

# Závěr

**V prvním odstavci závěru autor znovu uvede cíl práce** (bude shodný s cílem práce, který je uveden v úvodu).

I u závěru platí, že není zpravidla napsána na první pokus. Závěr je kapitolou, která by měla obsahovat zejména zhodnocení dosažených výsledků, hodnocení splnění cíle práce (nikoliv pouze konstatováním, že cíl práce byl/nebyl splněn; to skutečně nestačí), uvádí se také možnost uplatnění řešení v praxi apod.

V závěru by měl být stručně shrnut výsledek práce. Nejedná se však o konkrétní výsledky, které mají prostor v praktické části práce. Jedná se o výsledek shrnující v pár větách dosažení stanovených cílů. Autor při zadání práce čelí problému, který má vyřešit. Snadno se může stát, že během práce přijde na nové skutečnosti, se kterými na začátku nemusel počítat. Proto může uvést jisté odůvodnění, proč a jak v práci postupoval. Samozřejmě se nesmí autor od stanoveného cíle odchýlit výrazně. Jde o rozumnou toleranci přímo související s celou problematikou.

**Závěr představuje minimálně 1 normostranu textu (což odpovídá přibližně rozsahu tohoto obecného textu)**. Kromě výše uvedeného (zejména zhodnocení cílů práce) má obsahovat také shrnutí a zhodnocení teoretických poznatků (pokud jsou součástí práce), shrnutí a zhodnocení praktické části práce včetně přínosu pro praxi. Závěr také může obsahovat zdůvodnění nesplnění zadání práce, cíle práce apod.

Závěr se člení do odstavců. Nepatří sem nadpisy, obrázky, citáty (zvláště pak doslovné), číselné výsledky apod. Mělo by se jednat o čtivý a zajímavý text.

Osoby, které předem nebudou znát podrobnosti o práci (například stálí členové zkušební maturitní komise) by měli získat jasnou představu o práci po pročtení těchto kapitol: Úvod a cíl práce, Závěr. V kapitole Úvod a cíle práce čtenář načerpá obecné poznatky o dané práci. Po pročtení kapitoly Závěr by měl mít jasnou představu o tom, čím se měla práce zabývat, jak se podařilo cíle práce dosáhnout, jaký je přínos pro praxi apod.

**Seznam použitých zdrojů**

NOVÁK, Ondřej. *DoS útoky a jejich analýza*. Bakalářská práce. Slezská univerzita v Opavě, Filozoficko-přírodovědecká fakulta v Opavě. [online] 2023 [cit. 2024-12-28]. Dostupné z: https://is.slu.cz/th/uhwyf/.

Merkebaiuly, Medet. Overview of Distributed Denial of Service (DDoS) attack types and mitigation methods. InterConf. 10.51582/interconf.19-20.03.2024.048. 2024 [cit. 2024-12-14]. Dostupné také z: [https://www.researchga­te.net/publicati­on/379284307\_Over­view\_of\_Distributed\_Denial\_of\_Service\_DDoS\_attack\_types\_and\_mitigation\_methods](https://www.researchgate.net/publication/379284307_Overview_of_Distributed_Denial_of_Service_DDoS_attack_types_and_mitigation_methods).

Zargar, ST & Joshi, J & Tipper, D, A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks. IEEE Communications Surveys and Tutorials, 15 (4). 2046 – 2069. 2013 [cit. 2024-12-31]. Dostupné také z: [https://d-scholar­ship.pitt.edu/19225/1/FinalVersion.pdf](https://d-scholarship.pitt.edu/19225/1/FinalVersion.pdf)

Obaid, Hadeel. Denial of Service Attacks: Tools and Categories. International Journal of En­gineering Research and. V9. 10.17577/IJERTV9IS030289. (2020). Dostupné také z: [https://www.researchga­te.net/publication/341875337\_Denial\_of\_Service\_Attacks\_Tools\_and\_Categories](https://www.researchgate.net/publication/341875337_Denial_of_Service_Attacks_Tools_and_Categories).

Dandotiya, Abhinandan & Sharma, Palash & gole, Bharti & Dubey, Shruti & Dandotiya, Nidhi. An Empirical Analysis of DDoS Attack Detection and Mitigation Techniques: A Com­parative Review of Tools and Methods. International Journal of Scientific Research in Com­puter Science, Engineering and Information Technology. 10. 1099-1108. 10.32628/CSEIT2410462. (2024). ISSN: 2456-3307 Dostupné také z: [https://www.resear­chgate.net/publicati­on/386447490\_An\_Empiri­cal\_Analy­sis\_of\_DDoS\_At­tack\_Detecti­on\_and\_Mitigation\_Techniques\_A\_Comparative\_Review\_of\_Tools\_and\_Methods](https://www.researchgate.net/publication/386447490_An_Empirical_Analysis_of_DDoS_Attack_Detection_and_Mitigation_Techniques_A_Comparative_Review_of_Tools_and_Methods).

What is the low orbit ion cannon (LOIC)? Cloudflare. [online] 2024 [cit. 2024-12-27]. Do­stupné také z: <https://www.cloudflare.com/en-gb/learning/ddos/ddos-attack-tools/low-orbit-ion-cannon-loic/>

Hping3 Kali. [online] 2024 [cit. 2024-12-27]. Dostupné také z: [https://www.kali.org/to­ols/hping3/](https://www.kali.org/tools/hping3/)

Mukhadin Beschokov, What is HULK - HTTP Unbearable Load King?, wallarm [online] 2024 [cit. 2024-12-28]. Dostupné z: <https://www.wallarm.com/what/what-is-hulk-http-unbearable-load-king>

Netacad, CCNA: Enterprise Networking, Security, and Automation. [online] [Paywall] 2024 [cit. 2024-12-31]. Dostupné také z: <https://www.netacad.com/trainings/ccna-enterprise-networking-security-automation?courseLang=en-US>

Scarfone, K., & Mell, P. *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST Special Publication 800-94. National Institute of Standards and Technology. [online] (2007) [cit. 2024-12-31]. Dostupné také z: [https://tsapps.nist.gov/publica­tion/get\_pdf.cfm?pub\_id=50951](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50951)

F5, What Is a Load Balancer?. [online] 2023 [cit. 2024-12-31]. Dostupné také z: <https://www.f5.com/glossary/load-balancer>

Mirkovic, J., & Reiher, P. *A Taxonomy of DDoS Attack and DDoS Defense Mechanisms*. [online] (2004) [cit. 2024-12-31]. Dostupné také z: [https://www.prince­ton.edu/~rblee/ELE572Papers/Fall04Readings/DDoSmirkovic.pdf](https://www.princeton.edu/~rblee/ELE572Papers/Fall04Readings/DDoSmirkovic.pdf)

**Seznam příloh**

**Příloha č. 1: Název přílohy**

Samotná příloha. Nejčastěji se do příloh dávají např. návrhy plošných spojů, další obrazová dokumentace, ukázky kódu ve formě printscreenů apod. Pokud přílohy v práci nebudou, tuto kapitolu smažte.

**Příloha č. 2: Název přílohy**

[Příloha 1 Obsah named.conf.local 13](#_Toc192017249)

[Příloha 2 Obsah db.victim.com 13](#_Toc192017250)

[Příloha 3 Obsah db.1.16.176.in-addr.arpa 13](#_Toc192017251)

[Příloha 4 Bind shell script 14](#_Toc192017252)

[Příloha 5 Obsah bind.service 14](#_Toc192017253)

[Příloha 6 Hlavní script na CnC serveru 15](#_Toc192017254)

[Příloha 7 Service sniff scriptu na CnC serveru 16](#_Toc192017255)

[Příloha 8 Script na manuální spuštění útoku 16](#_Toc192017256)

[Příloha 9 Script – žádost o payload 17](#_Toc192017257)

[Příloha 10 Sniffer script na spuštění žádosti 18](#_Toc192017258)

**Seznam obrázků**

<https://www.svgrepo.com/collection/servers-isometric-icons/>

[Obrázek 1 Schéma základního DDoS útoku 5](#_Toc190713169)

[Obrázek 2 Schéma ICMP flood útoku 6](#_Toc190713170)

[Obrázek 3 Schéma TCP SYN flood útoku 7](#_Toc190713171)

[Obrázek 4 Schéma HTTP GET flood útoku 8](#_Toc190713172)

**Seznam tabulek** seznam obrázků.

**Seznam grafů atd.**

Viz seznam obrázků.