Střední průmyslová škola elektrotechnická  
a gymnázium V Úžlabině 320

Maturitní práce z odborných předmětů

Útoky typu DoS

Martin Smékal

I4.B

Vedoucí práce: Ing. David Čepička

Studijní obor: 18-20-M/01 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 17.03.2025 Martin Smékal

Podpis autora/autorky práce

Anotace

Tato práce se zaměřuje na útoky typu Denial of Service (DoS), které představují vážnou hrozbu pro dostupnost a stabilitu síťových služeb. Cílem práce je komplexně analyzovat principy a metody realizace DoS útoků, rozdělit je do jednotlivých kategorií (distribuované, volumetrické, protokolové a aplikační útoky) a prostřednictvím praktické části demonstrovat jejich reálný dopad v simulovaném prostředí.

Klíčová slova

Kyberbezpečnost, DoS, HTTP, TCP, Scapy, Hping3

Annotation

This work focuses on Denial of Service (DoS) attacks, which pose a serious threat to the availability and stability of network services. The aim of the study is to comprehensively analyze the principles and methods of executing DoS attacks, categorize them into different types (distributed, volumetric, protocol-based, and application-layer attacks), and demonstrate their real impact in a simulated environment through a practical approach.

Keywords

Cybersecurity, DoS, HTTP, TCP, Scapy, Hping3

Obsah

[1 Úvod a cíl práce 5](#_Toc193053218)

[2 Typy útoků DoS 6](#_Toc193053219)

[2.1 Princip a funkce útoků typu DoS 6](#_Toc193053220)

[2.2 Základní členění DoS útoků 6](#_Toc193053221)

[2.2.1 Distribuované útoky 6](#_Toc193053222)

[2.2.2 Volumetrické útoky 7](#_Toc193053223)

[2.2.3 Protokolové útoky 8](#_Toc193053224)

[2.2.4 Aplikační útoky 9](#_Toc193053225)

[2.3 Známé nástroje pro realizaci útoků 10](#_Toc193053226)

[2.3.1 LOIC 10](#_Toc193053227)

[2.3.2 Hping3 10](#_Toc193053228)

[2.3.3 HULK 10](#_Toc193053229)

[2.4 Prevence a obrana 11](#_Toc193053230)

[2.4.1 Firewall a IDS/IPS systémy 11](#_Toc193053231)

[2.4.2 Load balancery 11](#_Toc193053232)

[2.4.3 Protokoly 12](#_Toc193053233)

[2.4.4 Prevence ve firmách 12](#_Toc193053234)

[3 Simulace DDoS útoků 13](#_Toc193053235)

[3.1 Popis virtualizovaného prostředí 13](#_Toc193053236)

[3.2 Konfigurace prvků a služeb 14](#_Toc193053237)

[3.3 Příprava scénářů 16](#_Toc193053238)

[3.3.1 Útočník – CnC 16](#_Toc193053239)

[3.3.2 CnC – Botnet 16](#_Toc193053240)

[3.3.3 Ověření funkčnosti 19](#_Toc193053241)

[3.4 Scénář – HTTP flood 21](#_Toc193053242)

[3.5 Scénář – TCP SYN flood 25](#_Toc193053243)

[4 Závěr 27](#_Toc193053244)

[5 Seznam použitých zdrojů 28](#_Toc193053245)

1. Úvod a cíl práce

V dnešní digitální době, kdy jsou informační systémy a online služby nedílnou součástí každodenního života, se problematika kybernetické bezpečnosti stává stále naléhavější.

Tato práce se věnuje útokům typu Denial of Service (DoS), jež představují vážnou hrozbu pro dostupnost a stabilitu síťových služeb. V teoretické části jsou čtenáři přehledně představeny základní principy a metody realizace těchto útoků, přičemž jsou rozděleny do kategorií – distribuovaných, volumetrických, protokolových i aplikačních. Text se rovněž věnuje přehledu nástrojů, jako jsou LOIC, Hping3 či HULK, které slouží jak k testování, tak i k realizaci útoků.

Praktická část pak prostřednictvím simulovaného virtualizovaného prostředí demonstruje reálné scénáře útoků, konkrétně HTTP flood a TCP SYN flood, a představuje konfiguraci síťových zařízení, implementaci Command-and-Control serveru a fungování botnetu.

Celkově práce poskytuje určitý pohled na dynamiku DoS útoků, přičemž text je sestaven tak, aby byl přístupný i čtenářům bez hlubších odborných znalostí. Tento přístup má za cíl zvýšit povědomí o hrozbách, jež DoS útoky představují, a poukázat na efektivní postupy a technologická opatření k jejich mitigaci.

1. Typy útoků DoS

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

* 1. 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.

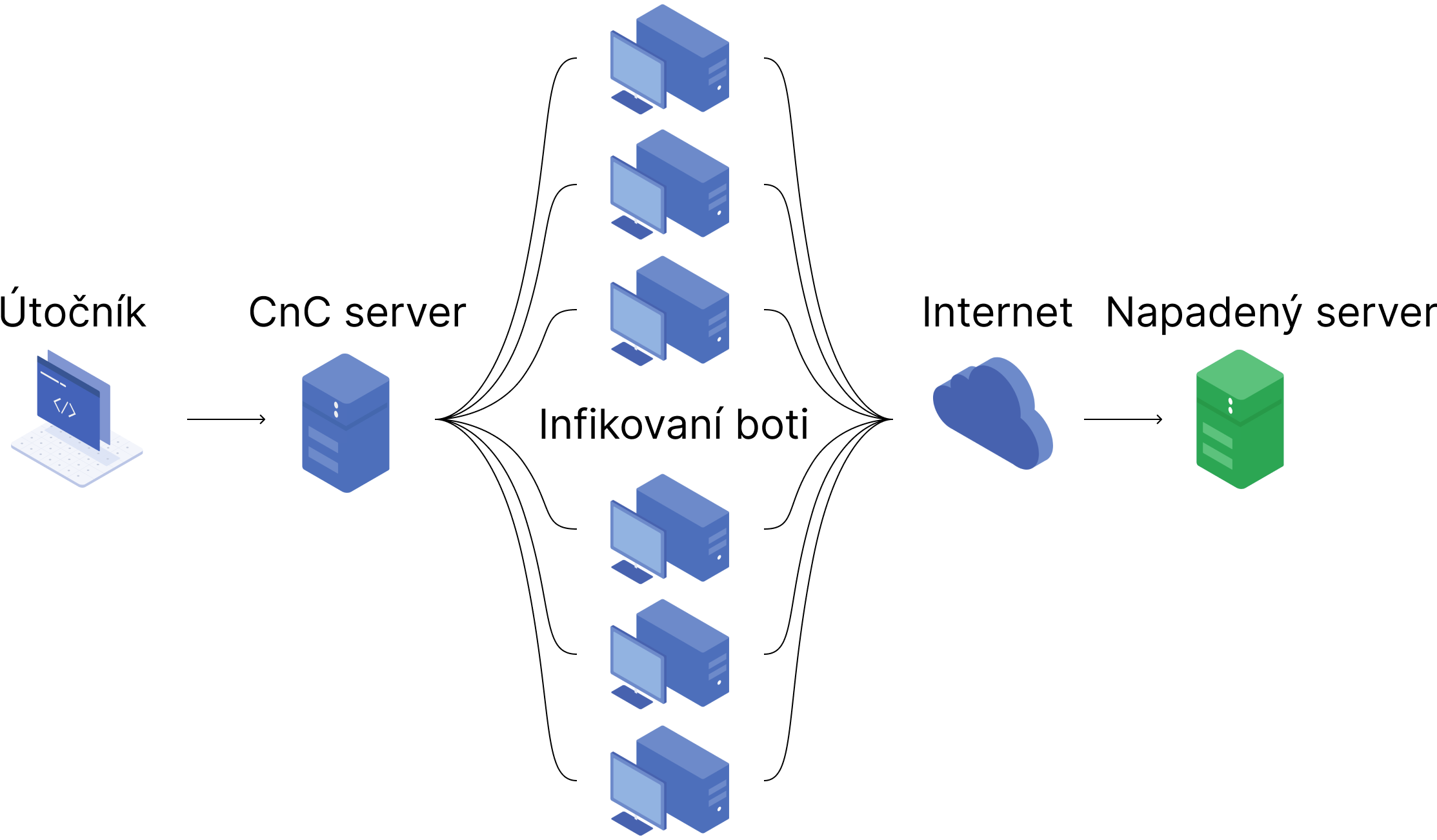
Většinou se u DoS útoků setkáváme se zaplavováním (flooding), což je odesílání velkého množství dat nebo požadavků s cílem přetížit šířku pásma (bandwidth) nebo výpočetní kapacitu cílového zařízení. Anebo také s využitím nějaké slabiny (exploit) v síťových protokolech nebo aplikacích.

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. Viz. Falowo a spol. 2024, nebo Kinter 2023 jako příklad rostoucího počtu útoků.

* 1. Základní členění DoS útoků
     1. Distribuované útoky

Distribuované útoky typu Denial of service, se liší od normálních DoS útoků tím, že odcházejí z více lokací najednou, a tak je účinnost útoků 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 Obrázku 1. (další obrázky topologie jsou navrženy podle vlastního zpracování s podporou volně dostupných svg obrázků dostupných z: https://www.svgrepo.com/collection/servers-isometric-icons/)



Obrázek 1. Schéma základního DDoS útoku – vlastní zpracování

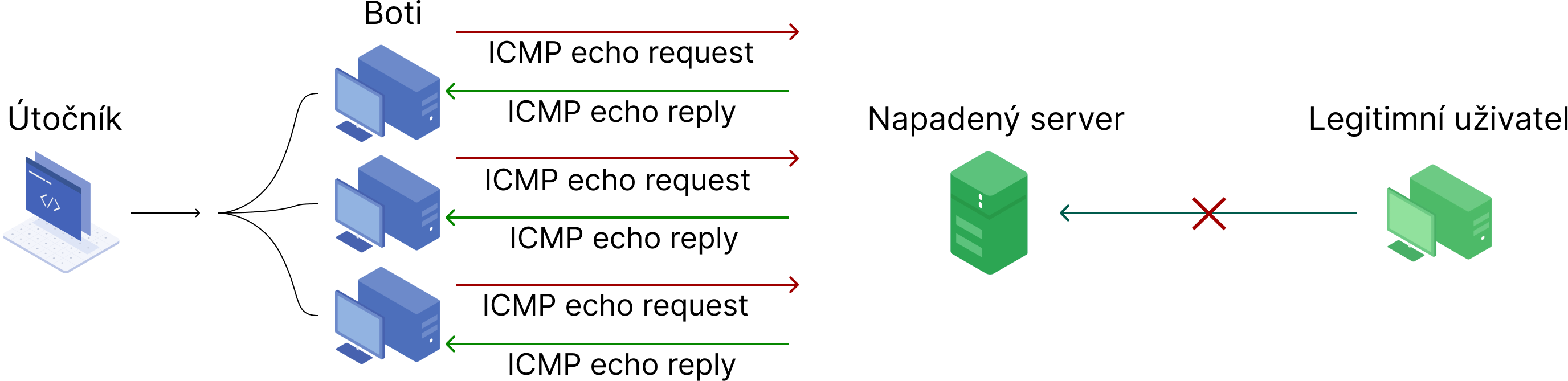
Na obrázku 1 vidíme použití strategie botnetu, což je kolekce botů (infikovaných počítačů), které jsou většinou pod hierarchickou kontrolou. Na konci této kontroly se obvykle vyskytuje také Command-and-Control (CnC) Server, který rozesílá útočníkovy instrukce svým botům.

* + 1. 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 mohly 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).



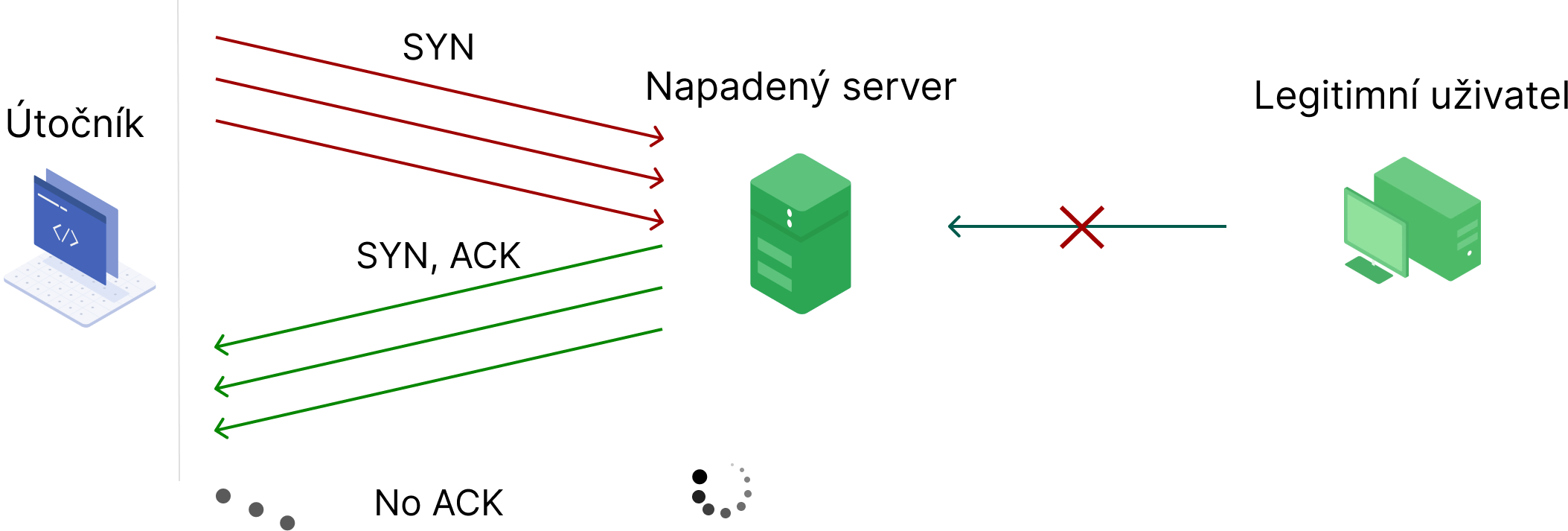
Obrázek 2. Schéma ICMP flood útoku – vlastní zpracování

Na levé straně Obrázku 2, který 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.

* + 1. 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 (Pps) paktetech za sekundu. Do těchto útoků pak spadá TCP SYN flood, Smurf útoky nebo útok Ping of death.



Obrázek 3. Schéma TCP SYN flood útoku – vlastní zpracování

TCP SYN flood funguje na základě již zmíněné 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 klientovy zprávy (ACK) a nakonec klient nazpět uzná serveru jeho zprávu (ACK).

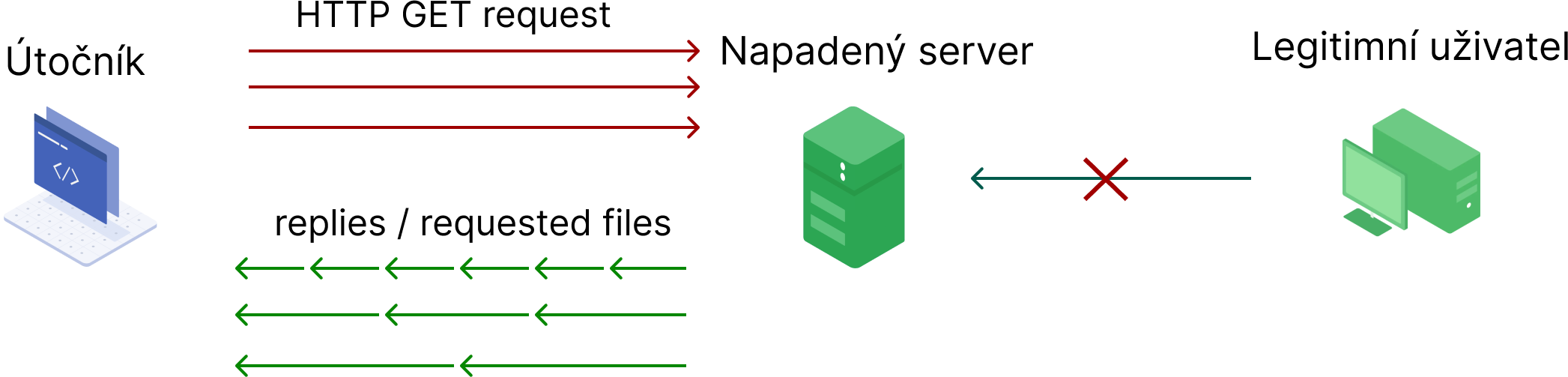
Trik v TCP SYN flood útoku je ten, že pokud klient neodpoví poslední zprávou (ACK), tak bude instance navázání komunikace stále otevřená. Nakonec server může dojít až do kritického bodu (viz. Obrázek 3), kdy bude přetížen a nebude schopen odpovídat na další TCP SYN zprávy.

* + 1. 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 4. Schéma HTTP GET flood útoku – vlastní zpracování

Na Obrázku 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.

* 1. Známé nástroje pro realizaci útoků
     1. 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)

* + 1. 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 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)

* + 1. 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 exploatace. 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)

* 1. 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)

* + 1. Firewall a IDS/IPS 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)

* + 1. 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)

* + 1. 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.

* + 1. Prevence ve firmách

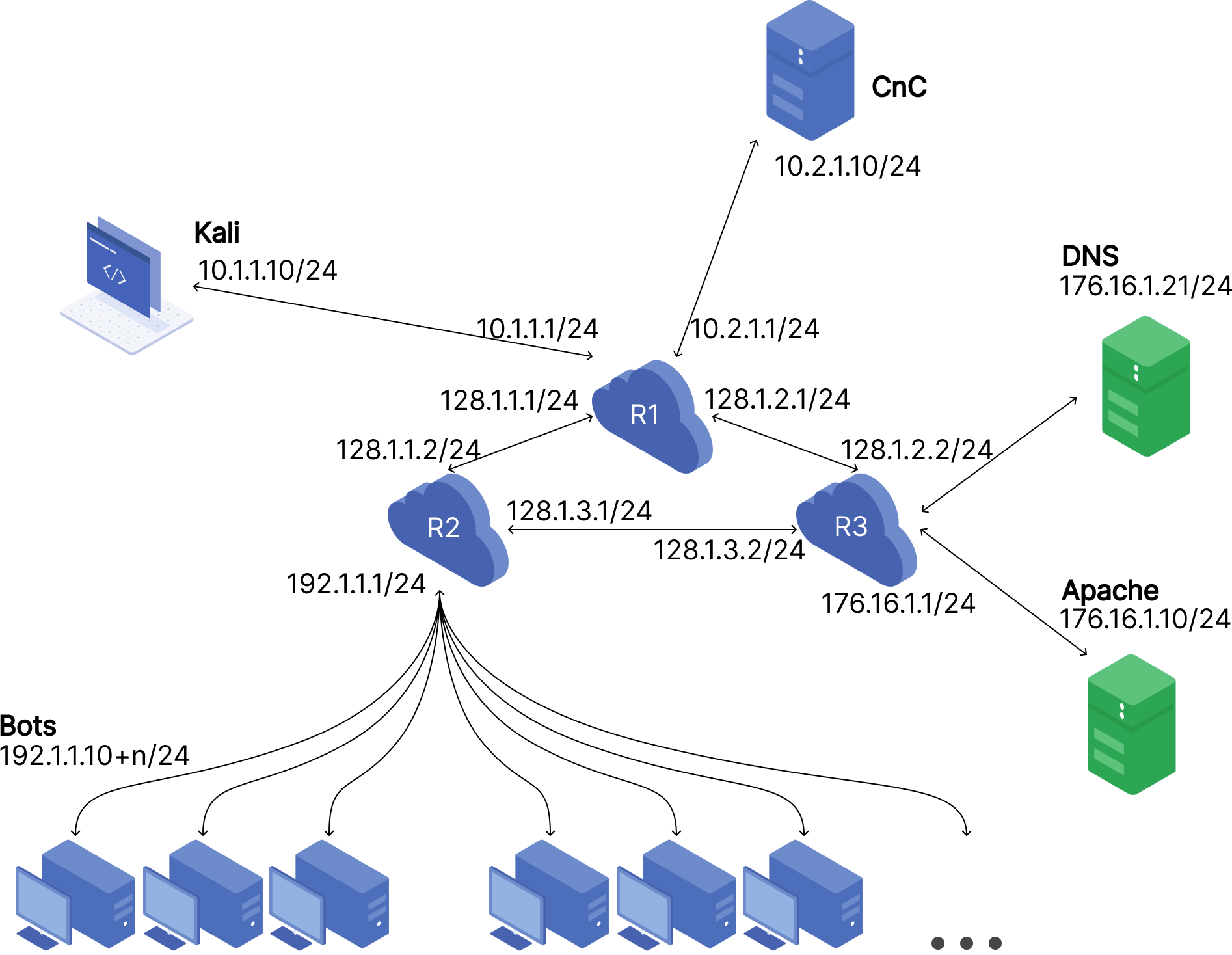
Prevence DoS a DDoS útoků ve firmách zahrnuje kombinaci technologických opatření, procesů a poučení zaměstnanců. Bezpečnostní opatření na síťové úrovni jsou například segmentování sítě či „geo-locking“, ale i ochranné systémy, jako už zmíněné Next-Generation Firewally a IPS. Dále se využívají i Webové Firewally (WAF) nebo API brány na ochranu API koncových zařízení před zneužitím nebo nadměrným zatěžováním.

Hlavní částí bezpečnosti ve firmách je však určitě i pravidelné školení zaměstnanců, plánování postupů na určité incidenty nebo pravidelné testování penetračními testy.

1. Simulace DDoS útoků

Tato praktická část se zaobírá dopadem DDoS útoků HTTP flood a TCP SYN flood. Řeší se celkový návrh virtualizovaného prostředí pro simulaci, postup pro zavedení CnC serveru a dalších prvků v celém scénáři, a nakonec jejich chování při probíhajícím útoku.

* 1. 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 má útočník 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.
* 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>).

* 1. Konfigurace prvků a služeb

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

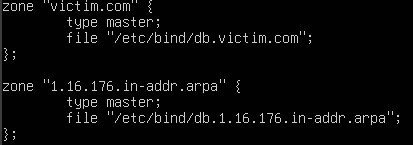
Router R1 s RouterOS má nastavený bridge pro rychlejší konfiguraci a setupování a je spravován pomocí programu WinBox. Na R1 je také nastavená src-nat NAT a DHCP client, což poskytuje komunikaci do internetu pro celou topologii.

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

U všech koncových zařízení se rozhraní konfigurovala za pomocí nástroje nmtui.

Na Apache 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 propsaly. Nakonec službu můžeme spustit pomocí „sudo /etc/init.d/apache2 start“, nebo „sudo systemctl start apache2.service“, záleží, jakou verzi operačního systému máme.

Konfigurace pro DNS server je zprovozněna podle služby bind9, takže hlavní soubory k úpravě jsou ve složce „/etc/bind/“. V „named.conf.options“ stačí změnit, na jaké IPv4 adrese budeme poslouchat a povolíme jakýkoliv provoz. V „named.conf.local“ (viz Obrázek 6) si už rozvrhneme pojmenování naší stránky (např. victim.com). Pak v samotné „named.conf“ ověříme, že oba soubory jsou zahrnuty.



Obrázek 6. 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 lze využít i stránky jako yoyo.org na generování SOA a NS záznamu. Konečná konfigurace našich zón bude vypadat tedy takto – Obrázky 7 a 8.



Obrázek 7. Obsah db.victim.com



Obrázek 8. Obsah db.1.16.176.in-addr.arpa

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“.

* 1. Příprava scénářů
     1. Útočník – CnC

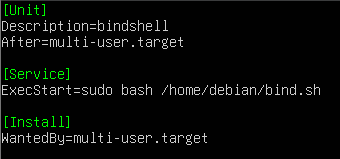
Ze zařízení útočníka se budeme připojovat na CnC server pomocí bind shell, za cílem stáhnout potřebný payload a spustit script, který řekne všem zaznamenaným botům, že si mají stáhnout payload a spustit ho.

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



Obrázek 9. Bind shell script

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



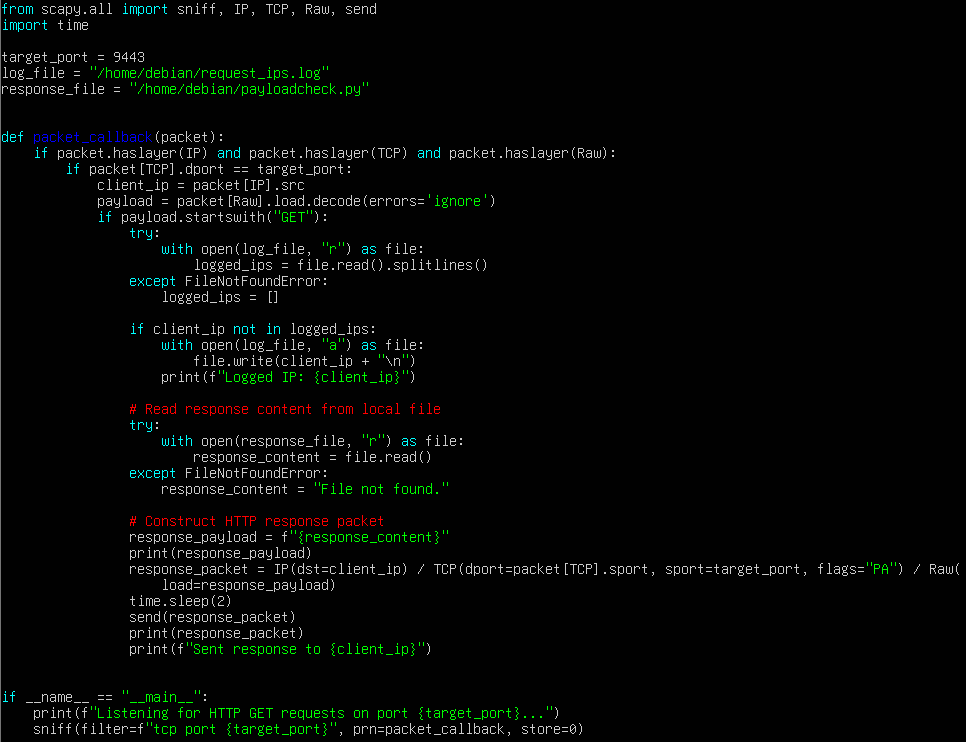
Obrázek 10. Obsah bind.service

Hlavní parametry v bind.service jsou tedy doba spuštění a umístění souboru pro spuštění. Čas spuštění nám udává „WantedBy=multi-user.target“, který se spustí na runlevelu 2,3,4 (tedy při spuštění zařízení). 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.

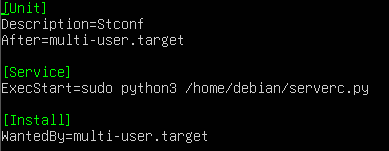
* + 1. CnC – Botnet

Další script (viz Obrázek 11), 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.



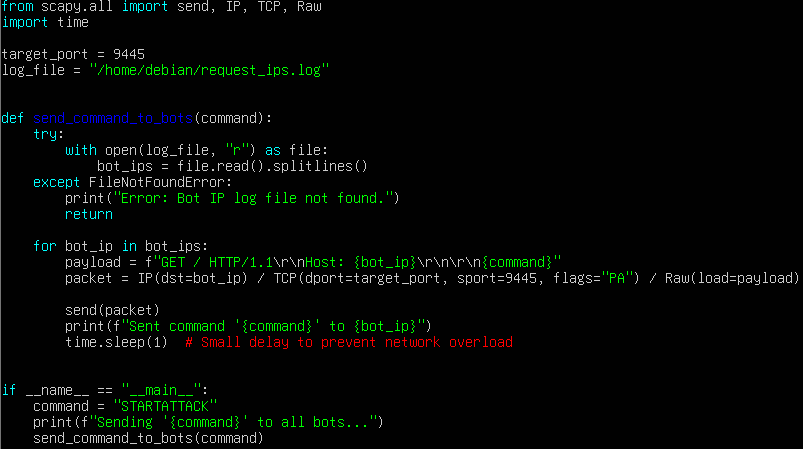
Obrázek 11. Hlavní script na CnC serveru

Pro posílání paketů se využívá balíček scapy. Celý script ověří, jestli je paket kompletní a porovná cílový port s tím, na kterém poslouchá. Pokud je paket typu GET, tak zaznamená jeho IP adresu do .log souboru. Načte payload soubor a na stejnou IP adresu a port pošle paket, ve kterém je obsažen náš škodlivý kód. Tento script běží neustále (běží jako .service (viz Obrázek 12)) a vlastně slouží jako „odchytáváč“ žádostí z botnetu.

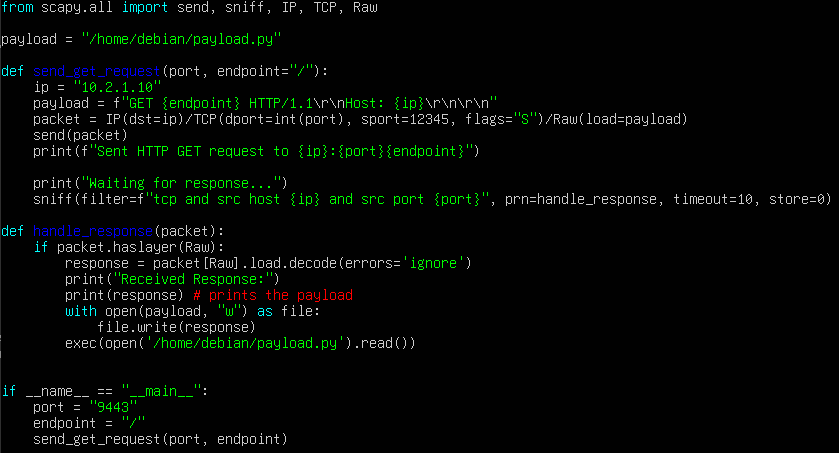


Obrázek 12. Service hlavního scriptu na CnC serveru

Poslední script, který je uložen na CnC serveru, je manuální „ping“ botnetu (viz Obrázek 13), 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.



Obrázek 13. Script na manuální spuštění útoku Botnet – Servery se službami

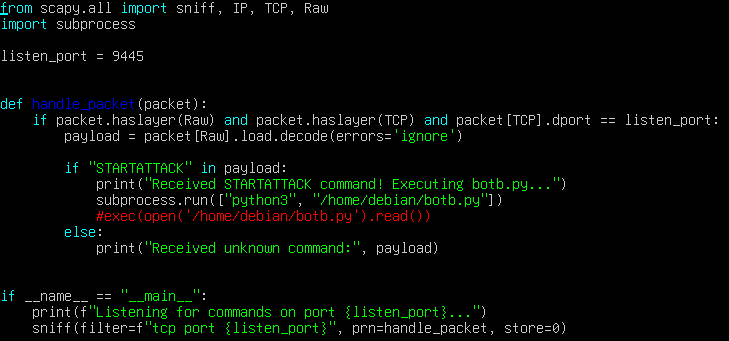


Obrázek 14. Script v botnetu – žádost o payload

Základní script (viz Obrázek 14), 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 Obrázku 13, což je zprostředkováno skrze sniffer script (viz Obrázek 15).

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 ji. Tím prakticky může spustit jakýkoliv script.

Dále musíme mít script na odposlech „pingu“ botnetu od CnC serveru, který nám tímto 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.



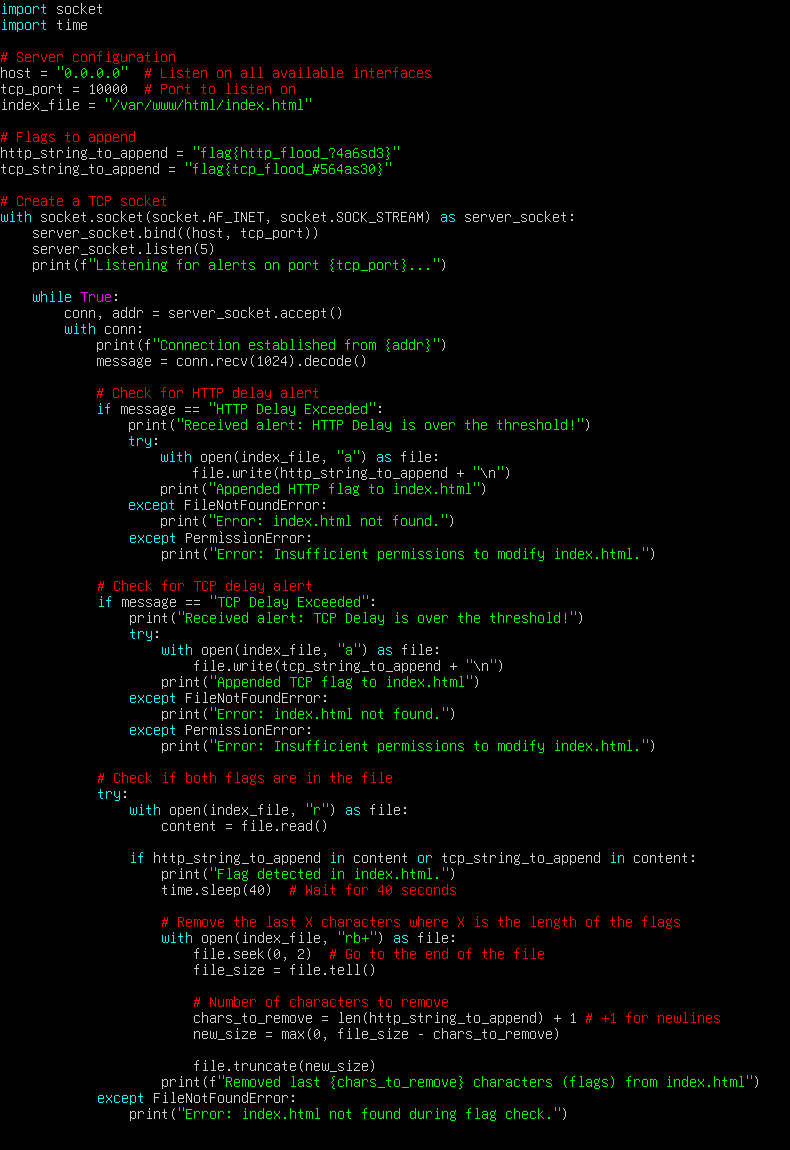
Obrázek 15. Sniffer script na spuštění žádosti

* + 1. Ověření funkčnosti

Ověření, zda útoky navyšují odezvu, můžeme zjistit buď přes ping na adresu serveru, nebo můžeme použít další kód na ověření, který nám rovnou na stránku může přidat náš řetězec za splnění dostatečné podmínky/úrovně odezvy.

Stačí nám tedy script, který bude odposlouchávat na portu, a pokud dostane zprávu, že odezva je vyšší než nastavená hodnota, tak přidá na konec index.html tajný řetěz (flag) (viz Obrázek 16).

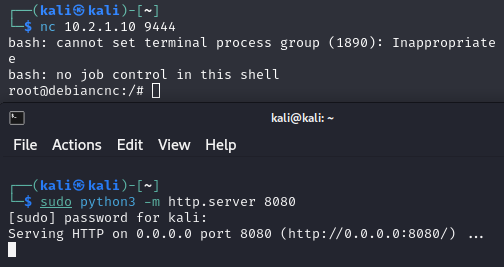
K němu na nevinném zařízení v topologii (pro ušetření místa můžeme klidně použít stroj útočníka) tedy poběží další script na kalkulaci a znázornění odezvy a následné poslání zprávy, že překročila hranici (viz Obrázek 23).



Obrázek 16. Listen script na Apache serveru pro řetězec

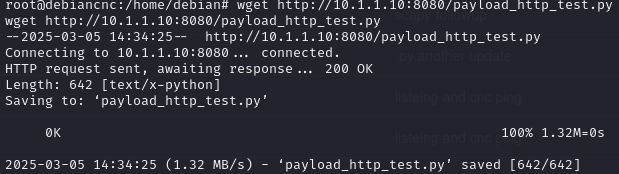
* 1. Scénář – HTTP flood

Scénář tedy začíná u útočníka (už tedy máme nakažené stroje, které se po spuštění zaznamenaly 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.



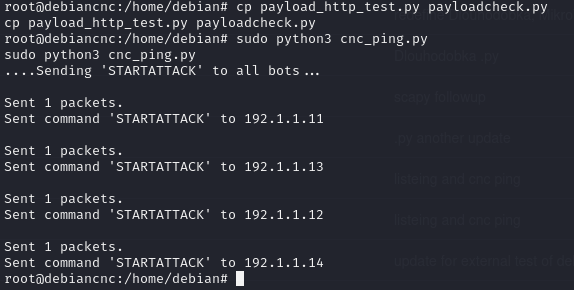
Obrázek 17. Bind shell + http server na Kali

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



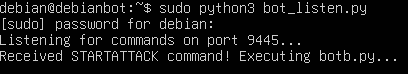
Obrázek 18. Wget http payload souboru

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



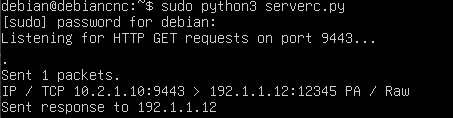
Obrázek 19. cnc\_ping-py

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



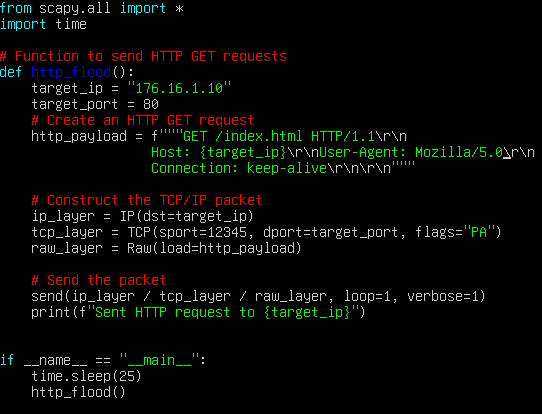
Obrázek 20. Přijetí zahájení útoku

Žádost se tedy dostane na CnC server, kde se zaznamená IP adresa a ověří se paket. Pak se pošle HTTP zprávou payload zpět kontaktovanému zařízení. Na Obrázku 21 je vidět, že CnC byl kontaktován botem s adresou 192.1.1.12 a jak odpovídá zpět.



Obrázek 21. Spuštění CnC scriptu

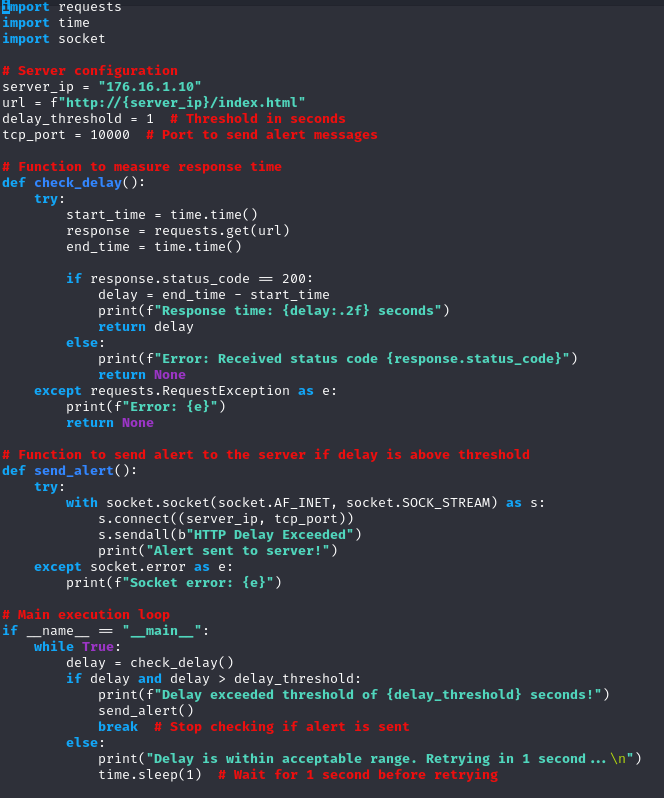
Jakmile bot obdrží paket, tak si jeho obsah uloží a uložený soubor spustí. V tomto scénáři použijeme tedy HTTP flood script (viz Obrázek 22).



Obrázek 22. Script na HTTP flood útok

Znovu 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(). Option Loop = 1 nám zajistí, že budeme posílat pakety, než se přístroj vypne, čímž se proces ukončí.

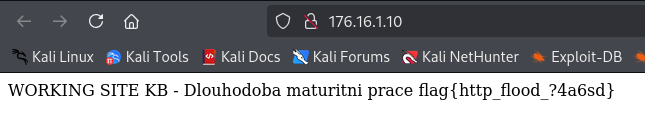
Na volitelném počítači (nejlépe Kali, kde se dá zobrazit stránka v prohlížeči) můžeme spustit script na měření odezvy (viz Obrázek 23). Ten nám po změření vypíše výslednou hodnotu, dále se koukne, jestli je odezva větší než nastavená hranice. Pokud ano, tak na Apache server pošle, že ji útok úspěšně překonal.



Obrázek 23. Script na měření odezvy

(Z mé zkušenosti stačí ~4 počítače v botnetu, aby se odezva začala pohybovat namísto ~0.01 okolo 0.5–2 sekundy. Více počítačů zapříčiní, že útok vyplýtvá výpočetní výkon (testováno na stroji s 32 GB RAM a 3.6 GHz i9 CPU) a server se z útoku dokáže docela dobře zotavit. Pro 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.)

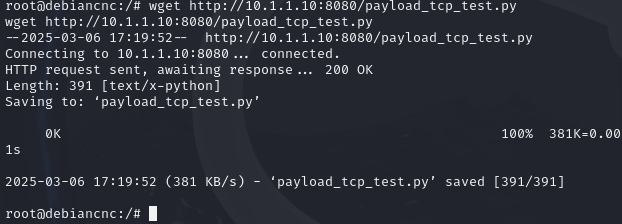
Následně po zaslání zprávy, kterou zpracuje script z Obrázku 16, se přidá na konec stránky příslušný řetězec, který se po určité době smaže, aby stránka nebyla plná řetězců pokud by se script spouštěl opakovaně.



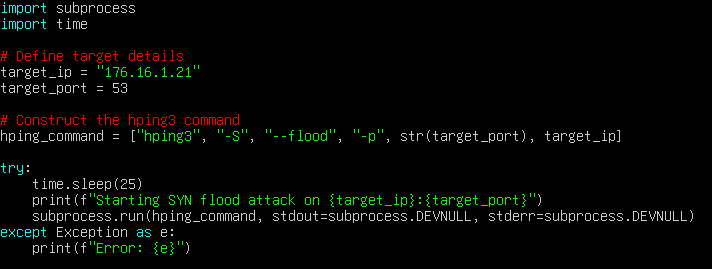
Obrázek 24. Splnění úlohy – HTTP řetězec

* 1. Scénář – TCP SYN flood

Druhý scénář používá stejnou topologii, ale jiný nástroj pro útok – hping3. Veliká změna to však v postupu nebude. Začátek je tedy stejný, akorát si stáhneme jiný script z Kali na CnC server (postupujeme tedy stejně se založením lokálního serveru a spustíme CnC „ping“ script).



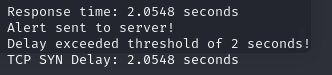
Obrázek 25. Wget tcp syn payload souboru



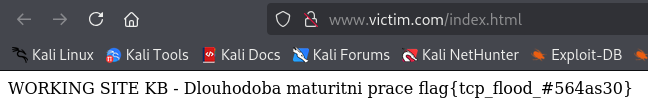
Obrázek 26. Script na TCP SYN flood

I když je obsah scriptu podobný, musíme u něj určit, jak bude vypadat hping3 příkaz. Nadeklarujeme si cílovou IP adresu a port a nastavíme možnosti u příkazu „-S“, jelikož chceme posílat SYN tcp flag a možnost „--flood“ pro nejrychlejší zasílaní paketů. Po spuštění útok zapříčiní, že DNS server na 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), 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.



Obrázek 27. Výstup scriptu na ověření odezvy



Obrázek 28. Splnění úlohy – TCP řetězec

1. Závěr

V závěru této práce jsou shrnuty klíčové poznatky získané z teoretické analýzy i praktické simulace útoků typu Denial of Service (DoS). Teoretická část poskytla ucelený přehled o různých metodách provádění útoků, včetně distribuovaných, volumetrických, protokolových a aplikačních útoků. Byly zde zdůrazněny mechanismy, které vedou k vyčerpání zdrojů cílového systému a následnému omezení nebo úplné nedostupnosti online služeb. Následný rozbor těchto útoků poukázal na jejich potenciál způsobit značné finanční a reputační škody, což podtrhuje důležitost dalšího výzkumu a vývoje obranných technologií.

Praktická část práce se zaměřila výhradně na stránku útoků a jejich simulaci ve virtualizovaném prostředí. Konkrétně byly implementovány a testovány útoky typu HTTP flood a TCP SYN flood, jejichž cílem bylo demonstrovat, jak lze přetížit cílový systém. Výsledky simulace prokázaly, že i relativně jednoduché útoky mohou způsobit zhoršení odezvy a destabilizaci cílové služby. Toto zjištění poukazuje na reálnou hrozbu, jež DoS útoky představují, a potvrzuje, že schopnost útoku narušit běžný provoz není zanedbatelná, zejména pokud by byl útok prováděn ve větším měřítku.

Dále získané poznatky o technických aspektech provádění útoků mohou sloužit jako základ pro budoucí vývoj bezpečnostních strategií. Znalost mechanismů útoků totiž umožňuje lépe navrhnout a implementovat ochranné systémy, které by dokázaly minimalizovat jejich dopad. Práce tedy nejen rozšiřuje teoretické poznatky v oblasti kybernetické bezpečnosti, ale také podněcuje další výzkum zaměřený na efektivní ochranu před DoS útoky. Celkově lze říci, že dosažené výsledky přispívají k lepšímu pochopení dynamiky útoků na online služby a upozorňují na nutnost nepřetržitého zvyšování bezpečnostních standardů v informačních systémech.

1. Seznam použitých zdrojů

Falowo, Olufunsho & Ozer, Murat & Li, Chengcheng & Bou abdo, Jacques. Evolving Malware & DDoS Attacks: Decadal Longitudinal Study. IEEE Access. PP. 1-1. 10.1109/ACCESS.2024.3376682. [online] 2024 [cit. 2024-12-14] Dostupné také z: https://www.researchgate.net/publication/378952021\_Evolving\_Malware\_DDoS\_Attacks\_Decadal\_Longitudinal\_Study

Kintr Lukáš. 2023 Report on the State of Cybersecurity in the Czech Republic. NÚKIB. [online] 2023. [cit. 2024-12-14] Dostupné také z: https://nukib.gov.cz/download/publications\_en/2023\_Report\_on\_the\_State\_of\_Cybersecurity\_in\_the\_Czech\_Republic.pdf

Merkebaiuly, Medet. Overview of Distributed Denial of Service (DDoS) attack types and mitigation methods. InterConf. 10.51582/interconf.19-20.03.2024.048. [online] 2024. [cit. 2024-12-14]. Dostupné také z: https://www.researchgate.net/publicati¬on/379284307\_Over-view\_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. [online] 2013. [cit. 2024-12-27]. Dostupné také z: https://d-scholar-ship.pitt.edu/19225/1/FinalVersion.pdf

Obaid, Hadeel. Denial of Service Attacks: Tools and Categories. International Journal of En-gineering Research and. V9. 10.17577/IJERTV9IS030289. [online] 2020. [cit. 2024-12-27] Dostupné také z: 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. [online] 2024. [cit. 2024-12-28] 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

What is the low orbit ion cannon (LOIC)? Cloudflare. [online] 2024 [cit. 2024-12-28]. Do-stupné také z: https://www.cloudflare.com/en-gb/learning/ddos/ddos-attack-tools/low-orbit-ion-cannon-loic/

Hping3. Kali. [online] 2024 [cit. 2024-12-28]. Dostupné také z: https://www.kali.org/to-ols/hping3/

Mukhadin Beschokov, What is HULK - HTTP Unbearable Load King?, wallarm [online] 2024 [cit. 2024-12-29]. 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/publica-tion/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.prince-ton.edu/~rblee/ELE572Papers/Fall04Readings/DDoSmirkovic.pdf

Seznam obrázků

[Obrázek 1. Schéma základního DDoS útoku – vlastní zpracování 7](#_Toc192950958)

[Obrázek 2. Schéma ICMP flood útoku – vlastní zpracování 8](#_Toc192950959)

[Obrázek 3. Schéma TCP SYN flood útoku – vlastní zpracování 8](#_Toc192950960)

[Obrázek 4. Schéma HTTP GET flood útoku – vlastní zpracování 9](#_Toc192950961)

[Obrázek 5. Topologie praktické části – vlastní zpracování 13](#_Toc192950962)

[Obrázek 6. Obsah named.conf.local 15](#_Toc192950963)

[Obrázek 7. Obsah db.victim.com 15](#_Toc192950964)

[Obrázek 8. Obsah db.1.16.176.in-addr.arpa 15](#_Toc192950965)

[Obrázek 9. Bind shell script 16](#_Toc192950966)

[Obrázek 10. Obsah bind.service 16](#_Toc192950967)

[Obrázek 11. Hlavní script na CnC serveru 17](#_Toc192950968)

[Obrázek 12. Service hlavního scriptu na CnC serveru 17](#_Toc192950969)

[Obrázek 13. Script na manuální spuštění útoku Botnet – Servery se službami 18](#_Toc192950970)

[Obrázek 14. Script v botnetu – žádost o payload 18](#_Toc192950971)

[Obrázek 15. Sniffer script na spuštění žádosti 19](#_Toc192950972)

[Obrázek 16. Listen script na Apache serveru pro řetězec 20](#_Toc192950973)

[Obrázek 17. Bind shell + http server na Kali 21](#_Toc192950974)

[Obrázek 18. Wget http payload souboru 21](#_Toc192950975)

[Obrázek 19. cnc\_ping-py 22](#_Toc192950976)

[Obrázek 20. Přijetí zahájení útoku 22](#_Toc192950977)

[Obrázek 21. Spuštění CnC scriptu 22](#_Toc192950978)

[Obrázek 22. Script na HTTP flood útok 23](#_Toc192950979)

[Obrázek 23. Script na měření odezvy 24](#_Toc192950980)

[Obrázek 24. Splnění úlohy – HTTP řetězec 25](#_Toc192950981)

[Obrázek 25. Wget tcp syn payload souboru 25](#_Toc192950982)

[Obrázek 26. Script na TCP SYN flood 25](#_Toc192950983)

[Obrázek 27. Výstup scriptu na ověření odezvy 26](#_Toc192950984)

[Obrázek 28. Splnění úlohy – TCP řetězec 26](#_Toc192950985)