Princípy informačnej bezpečnosti

Špecifikácia projektu

Zvýšenie odolnosti webových aplikácií proti útokom typu DDoS, za pomoci horizontálneho škálovania

Miroslav Hájek

Akademický rok: 2020 / 2021

Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave

Vymedzenie predmetnej domény

Útoky odmietnutia služby

Pozornosť projektu bude sústredená hlavne na aspekt dostupnosti ako jeden z kľúčových prvkov informačnej bezpečnosti. Pozrieme sa na hrozbu útokov typu odmietnutia služby (Denial of Service) zadelených podľa vektorov útoku. Určite neprehliadneme ani ich distribuovanú realizáciu prostredníctvom botnetov, nakoľko oplývajú potenciálne významnou ničivou silou v kybernetickom priestore. Preskúmame stratégie predchádzania voči vyskytujúcim sa rizikám príkladmi vhodnej konfigurácie infraštruktúry. Následne objasníme spôsoby odhalenia útokov monitorovaním kľúčových metrík cez Zabbix a dekompozíciou záznamov z logov.

Ochrana a redundancia aplikačných serverov

Popísaná prevencia bude spočívať v predstavení odlišných techník obrany proti škodlivo nadmernej sieťovej premávke na rôznych vrstvách sieťového modelu RM OSI. Pre úplnosť letmo spomenieme tiež ochranu aktívnymi sieťovými prvkami. Prioritne sa však budeme venovať vyvažovaniu záťaže (Load balancing) medzi viaceré redundantné servery s horizontálne škálovanou webovou aplikáciou spojeného s využitím rate limiting. Pričom cieľom bude dosiahnuť vyššiu spoľahlivosť a odolnosť proti výpadkom (Failover).

Na aplikačnej vrstve porovnáme starší prístup vyvažovania záťaže cez DNS záznamy a súčasné obľúbené open-source softvérové riešenia reverznej proxy a load balancera: HAProxy a NGINX. Pri simultánnej prevádzke viacerých inštancií aplikácie vzniká problém so zachovaním perzistentných relácií (s cookies). V náväznosti na to, popíšeme rozdiely medzi dostupnými algoritmami vyvažovania záťaže a v situáciach s bezstavovým a stavovým spojením.

Riadenie prístupu na stroje za reverznou proxy

Okrem pohľadu na nasadenú webovú aplikáciou predstavujúcu dopytovanú službu ako celok je nutné brať na zreteľ ochranu jednotlivých serverov, ktoré využívame na zabezpečenie kontinuálnej dostupnosti. Pozrieme sa na riadenie administrátorského prístupu cez ACL podporeného voľbou silných hesiel a znemožnením ich skúšania útočníkom alebo zavedením TLS. Zároveň je potrebné poohliadať sa na nebezpečie plynúce z prílišnej transparentnosti ohľadom používaných systémov, ktoré môžu vynášať hlavičky protokolu HTTP a uľahčiť tým nájdenie zraniteľností hackerom.

Tématické rozčlenenie problematiky

Zo zamerania vyplývajú nasledujúce tématické sekcie, s ktorými je asociovaná príslušná nosná terminológia, techniky, nástroje, perspektívne na prejednanie v projekte:

1. Dostupnosť ako bezpečnostný atribút

Význam elementu dostupnosti informácií ako piliera informačnej bezpečnosti (CIA triáda). Motivácia a príležitosti na kybernetický útok DDoS z kriminalistického hľadiska. Objasnenie výberu často vytipovaných obetí. Zdroje: [1], [2].

2. Anatómia útokov Denial of Service

Najčastejšie cielené vektory útoku pre hrozbu odmietnutia služby: zahltenie linky, zaplnenie stavových tabuliek, vyčerpanie výpočtových zdrojov. Formy realizácie najmä v podobe botnetov ovládaných cez C&C. Taxonómia DDoS útokov rozlíšených do troch hlavných kategórii s analýzou konkrétnych zraniteľností: veľkoobjemové (volumetrické) útoky s podskupinou amplifikačných reflexívnych útokov, protokolové a aplikačné útoky. Početnosť výskytov takýchto hrozieb v nedávnom období a rola aktívnych sieťových zariadení v usmernení nadmernej premávky: black hole filtering, IP black lists, NAT s ACL. Zdroje: [3], [4], [5], [6], [7], [8].

3. Škálovanie webových aplikácií

Princíp a dopad horizontálneho a vertikálneho škálovania aplikácie dostupnej cez počítačovú sieť. Pri vertikálnom škálovaní si budeme všímať výsledky navýšenia počtu procesov alebo vlákien u Dispatcher-Worker modelu. Horizontálnym škálovaním vytvoríme symetrizáciou zátaže pomedzi viaceré inštancie. Podľa schopností vrstiev sieťového modelu rozlišujeme agregáciu liniek (*bonding, trunking*) na spojovej vrstve (L2), viaccestné smerovanie protokolom ECMP (*Equal-cost multi-path routing*) na sieťovej vrstve (L3) a reverznú proxy na nadviazanie spojení na transportnej (L4 - protokol TCP) a aplikačnej vrstve (L7 - protokol HTTP). Doplníme prístupom riešiacim load balancing DNS záznamami (nástroje: *dnsmasq*, *bind9*). Zdroje: [9], [10], [11], [12], [13].

4. Algoritmy vyvažovania záťaže

Požiadavky prichádzajúce na proxy server musia byť v distribuovanom prostredí pridelené na práve jeden cieľový webový server, ktorý ich vybaví a vráti odpoveď. Podľa nárokov na výkon a správanie sa jednotlivých komponentov systému existujú viaceré stratégie výberu destinácie. Rozlišujeme medzi technikami statického a dynamického vyvažovania záťaže. Medzi statické algoritmy patria *náhodné pridelenie*, *round-robin*, *najmenej pripojení* a ich vážené varianty, *konzistentné hašovanie* IP adresy zdroja alebo URI cieľa. Dynamické stratégie pridávajú komplexnosť v tom, že uzol rozdeľujúci požiadavky musí pravidelne zbierať údaje o celkovom stave ním spravovaných serverov. Majorita webstránok musí uchovávať stav v cookies a load balancer sa musí tomu podriadiť: riešením sú tzv. *sticky sessions*. Zdroje: [14], [15], [16].

5. Softvérové nástroje na vyvažovanie záťaže

Porovnáme funkčné, výkonnostné a bezpečnostné možnosti dvoch populárnych load balancerov HAProxy a NGINX. Predvedieme ukážky ich základnej konfigurácie a efekt rôznych algoritmov na vyvažovanie horizontálne škálovanej aplikácie, ktorá využíva webový server NGINX. Pôjde jednak o jednoduchú vlastnú webstránku galérie so statickým obsahom, ktorá bude v ďalšom porovnávaní rozšírená o dynamickú časť v jazyku PHP (cez FPM), s prihlasovaním na personalizovaný výber fotografií na titulnej stránke. Pri nastavovaní load balancerov proti najčastejším DDoS útokom sa tiež oprieme o odporúčania od tvorcov týchto nástrojov na prevenciu spoliehajúcu sa predovšetkým na limitovaní počtu aktívnych spojení alebo frekvencie požiadaviek na jedného klienta cez algoritmus *leaky bucket* a podobne. Zdroje: [17], [18], [19], [20], [21], [**haproxy-php-fpm**], [22], [23].

6. Vysoká dostupnosť služieb

Nasadením load balancera síce dosiahneme redundanciu pre webový server, ale vyrobili sme centrálny bod zlyhania, ktorý eliminujeme duplikáciou *HAProxy* do klastra na dvoch strojoch v konfigurácii Active-Passive nástrojom *Keepalived*. Dosiahne toho priradením jednej virtuálnej IP adresy pre klaster a prepnutím na zálohu (Failover) po signalizovaní BFD (Bidirectional Forwarding Detection) prostredníctvom protokolu VRRP (Virtual Router Redundancy Protocol). Zdroje: [24].

7. Ochrana webových serverov

Protokol HTTP dokáže v hlavičkách odpovedí (Forwarded, Server, atď.) odkryť vlastnosti používaných systémov a dokonca adresy serverov skrytých za reverznou proxy. Tým môže odhaliť útočníkovi priamy prístup na jednotlivé servery alebo koncové body, ak nie sú chránené napr. ACL. Zabezpečenie spojenia s aplikáciou s TLS vložením X.509 certifikátov a proces prihlasovania do systému cez HTTP(S). Zdroje: [25], [26], [27], [28]

8. Monitorovanie webovej aplikácie

Sledovanie stavu sieťovej premávky ako sú vyťaženosť linky, nadviazané spojenia a ich časová distribúcia a tiež kontrolovanie utilizácie výpočtových zdrojov, ktoré zabezpečujú dostupnosť služby. Reaktívna analýza útokov z loggov load balancera a webového servera v Common Logfile format. Automatizácia upozornení na prípadné hrozby v monitorovacom nástroji Zabbix integrovanom rozšíreniami s demonštrovanými load balancermi. Preskúmanie najdôležitejších metrík, z tých ktoré sú k dispozícii a kontrola ich vývoja počas DoS. Zdroje: [29], [30], [31], [32]

9. Simulácia útokov a záťažové testy

Výsledky záťažových testov vlastnej webovej aplikácie spustenej v Docker kontajneroch cez *docker compose* a na fyzických zariadeniach Raspeberry Pi (2x Pi 3B, 1x Pi 2B). Účinky existujúcich nástrojov na stress testing: *ab, t50, thc-ssl-dos, SlowHTTPTest, High Orbit Ion Cannon*, ale aj vlastnými viacvláknovými skriptami na UDP flood, SYN flood, Slow Loris. Pokus o vydávanie sa za rôzne počítače technikou IP Spoofing. Vyhodnotenie dopadov pre rôzny počet bežiacich inštancií web servera, počet workerov a nastavenia load balancingu.

Časový plán čiastkových cieľov

- 4. týždeň: Pripravená testovacia architektúra pre load balancing a vysokú dostupnosť v Docker kontaineroch a na Raspberry Pi, spolu so statickou web stránkou a rozpracovaným prihlasovaním. Sprevádzkovaný Zabbix s rozšíreniami pre HAProxy a NGINX.
- 8.týždeň: Zrealizované vybrané DoS útoky a záťažové testy zvolenými nástrojmi aj vlastnými skriptami a kvantifikované výsledky. Na teoretickej úrovni rozobrané nosné témy práce, okrem sekcií: škálovanie webových aplikácii, monitoring aplikácie a implementácia TLS a ACL pre web servery, dopad stavových spojení na load balancing

Zoznam relevantných zdrojov pre rešerš

1. MIR, Suhail; QUADRI, Syed. Information Availability: An Insight into the Most Important Attribute of Information Security. *Journal of Information Security*. 2016, vol. 07, pp. 185–194. Available from DOI: 10.4236/jis.2016.73014.
2. ABHISHTA, Abhishta; HEESWIJK, Wouter van; JUNGER, Marianne; NIEUWENHUIS, Bart; JOOSTEN, Reinoud. Why would we get attacked? An analysis of attacker's aims behind DDoS attacks. *Journal of Wireless Mobile Networks*. 2020, vol. 11, pp. 3–22. Available from DOI: 10.22667/JOWUA.2020.06.30.003.
3. MIRKOVIC, Jelena; REIHER, Peter. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*. 2004, vol. 34, no. 2, pp. 39–53. Available from DOI: 10.1145/997150.997156.
4. ALOMARI, Esraa; MANICKAM, Selvakumar; B. GUPTA, B.; KARUPPAYAH, Shankar; ALFARIS, Rafeef. Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art. *International Journal of Computer Applications*. 2012, vol. 49, no. 7, pp. 24–32. Available from DOI: 10.5120/7640-0724.
5. *DDoS Handbook: The Ultimate Guide to Everything You Need to Know about DDoS Attacks*. Radware, 2016. https://www.radware.com/getattachment/Security/Research/702/Radware_DDoS_Handbook_2015.pdf.aspx.
6. ENISA Threat Landscape 2020 - Distributed denial of service. *ENISA*. 2020. https://www.enisa.europa.eu/publications/enisa-threat-landscape-2020-distributed-denial-of-service.
7. PRASAD, K.Munivara; REDDY, A.Rama Mohan; RAO, K.Venugopal. DoS and DDoS Attacks: Defense, Detection and Traceback Mechanisms -A Survey. *Global Journal of Computer Science and Technology (GJCST)*. 2014, vol. 14. ISSN 0975-4172. https://core.ac.uk/reader/231149430.
8. *Ochrana pred útokmi DDoS (Príručka administrátora)*. CSIRT.SK, 2013. https://www.csirt.gov.sk/doc/DDoS_CSIRT.pdf.

9. *IEEE 802.3: Tutorial on Link Aggregation and Trunking*. 1997. https://grouper.ieee.org/groups/802/3/trunk_study/tutorial/index.html.
10. THALER, D.; HOPPS, C. *Multipath Issues in Unicast and Multicast Next-Hop Selection* [Internet Requests for Comments]. RFC Editor, 2000-11. RFC, 2991. RFC Editor. ISSN 2070-1721. http://www.rfc-editor.org/rfc/rfc2991.txt.
11. HOPPS, C. *Analysis of an Equal-Cost Multi-Path Algorithm* [Internet Requests for Comments]. RFC Editor, 2000-11. RFC, 2992. RFC Editor. ISSN 2070-1721. http://www.rfc-editor.org/rfc/rfc2992.txt.
12. BOURKE, Tony. *Server load balancing*. O'Reilly, 2001.
13. SOUNDARABAI, Paulsingh; SANDHYA, Rani; SAHAI, Ritesh; K R, Venugopal; PATNAIK, Lalit. COMPARATIVE STUDY ON LOAD BALANCING TECHNIQUES IN DISTRIBUTED SYSTEMS. 2012, vol. 6.
14. MCHEICK, Hamid; ALARDAWY, Zaid; LAKISS, Abbass. Evaluation of Load Balance Algorithms. In: 2011, pp. 104–109. Available from DOI: 10.1109/SERA.2011.46.
15. ROY, Saumendu; HOSSAINA, Dr. Md. Alam; SEN, Sujit Kumar; HOSSAIN, Nazmul; ASIF, Md. Rashid Al. Measuring the Performance on Load Balancing Algorithms. *Global Journal of Computer Science and Technology*. 2019, vol. 19. ISSN 0975-4172.
16. *Sticky Session*. 2020. https://www.imperva.com/learn/availability/sticky-session-persistence-and-cookies/.
17. *NGINX Load Balancer Guide*. 2021. https://docs.nginx.com/nginx/admin-guide/load-balancer/.
18. DEJONGHE, Derek. In: *NGINX cookbook: advanced recipes for high performance load balancing*. O'Reilly, 2019, pp. 9–25.
19. *HAProxy documentation*. 2021. http://cbonte.github.io/haproxy-dconv/.
20. ASSMANN, Baptiste. Use a Load Balancer as a First Row of Defense Against DDOS. *HAProxy Blog*. 2012. https://www.haproxy.com/blog/use-a-load-balancer-as-a-first-row-of-defense-against-ddos/.
21. LAVOLE, Chad. Application-Layer DDoS Attack Protection with HAProxy. *HAProxy Blog*. 2018. https://www.haproxy.com/blog/application-layer-ddos-attack-protection-with-haproxy/.
22. *Mitigating DDoS Attacks with NGINX and NGINX Plus*. 2015. https://www.nginx.com/blog/mitigating-ddos-attacks-with-nginx-and-nginx-plus/.
23. RAMIREZ, Nick. *HAProxy Rate Limiting: Four Examples*. 2019. https://www.haproxy.com/blog/four-examples-of-haproxy-rate-limiting/.
24. *Keepalived User Guide*. 2021. https://keepalived.readthedocs.io/en/latest/.
25. *Linux Hardening (Príručka administrátora)*. CSIRT.SK, 2013. https://www.csirt.gov.sk/doc/Hardened_v1.pdf.
26. *Kontrolný zoznam pre bezpečnosť webových aplikácií*. CSIRT.SK, 2020. https://www.csirt.gov.sk/doc/Checklist2.0.pdf.
27. FIELDING, Roy T.; GETTYS, James; MOGUL, Jeffrey C.; NIELSEN, Henrik Frystyk; MASINTER, Larry; LEACH, Paul J.; BERNERS-LEE, Tim. *Hypertext Transfer Protocol – HTTP/1.1* [Internet Requests for Comments]. RFC Editor, 1999-06. RFC, 2616. ISSN 2070-1721. http://www.rfc-editor.org/rfc/rfc2616.txt.
28. *HTTP headers reference*. 2021. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers.
29. *Log Files*. 2021. https://httpd.apache.org/docs/trunk/logs.html.
30. MHEDHBI, Moemen. *Introduction to HAProxy Logging*. 2019. https://www.haproxy.com/blog/introduction-to-haproxy-logging/.
31. *Zabbix Nginx integration*. 2021. https://www.zabbix.com/integrations/nginx.
32. *Zabbix HAProxy integration*. 2021. https://www.zabbix.com/integrations/haproxy.