Praca inżynierska

System monitoringu serwerów „ServerMonitor”

Autor: Konrad Komada

Promotor: dr Beata Pańczyk

Cel pracy

- Zaprojektowanie zbioru skryptów i aplikacji automatyzujących czynności związane z monitoringiem stanu serwerów, ich zasobów, usług, parametrów pracy oraz wysyłających powiadomienia w przypadku pojawienia się anomalii, mogących skutkować awarią oraz w przypadku zaistnienia awarii.
- Skrypty powinny cechować się dużą autonomicznością działania, czyli powinny wykonywać wszelkie możliwe czynności zaradcze i naprawcze automatycznie.
- Cały system powinien być łatwo skalowalny, czyli powinien umożliwiać dokładanie nowych usług i serwerów do monitoringu.

Zakres pracy

- Skrypty i aplikacja przeznaczone do monitoringu serwerów pracujących pod kontrolę systemu Linux i jego usług.
- Główną technologią skryptową jest język Bash oraz Sh.
- Technologią dla części aplikacyjnej jest język C++ oraz biblioteka QT.
- Powiadomienia drogą e-mail oraz przedstawi podstawowe parametry serwera poprzez graficzny interfejs użytkownika (GUI).
- Wykorzystana technologie sieciowe TCP/IP (ipv4 oraz ipv6).

Architektura systemu

- Architektura jedno i dwuwarstwowa dla skryptów.
- Architektura dwuwarstwowa dla aplikacji - Klient-Serwer.
- Środowisko Linux Debian 7.8 (Wheezy) na serwerze VPS z oprogramowaniem DirectAdmin (panel serwera).
- Środowisko Linux lub Cygwin po stronie klienta.
- Języki programowania BASH, SH, QT C++.

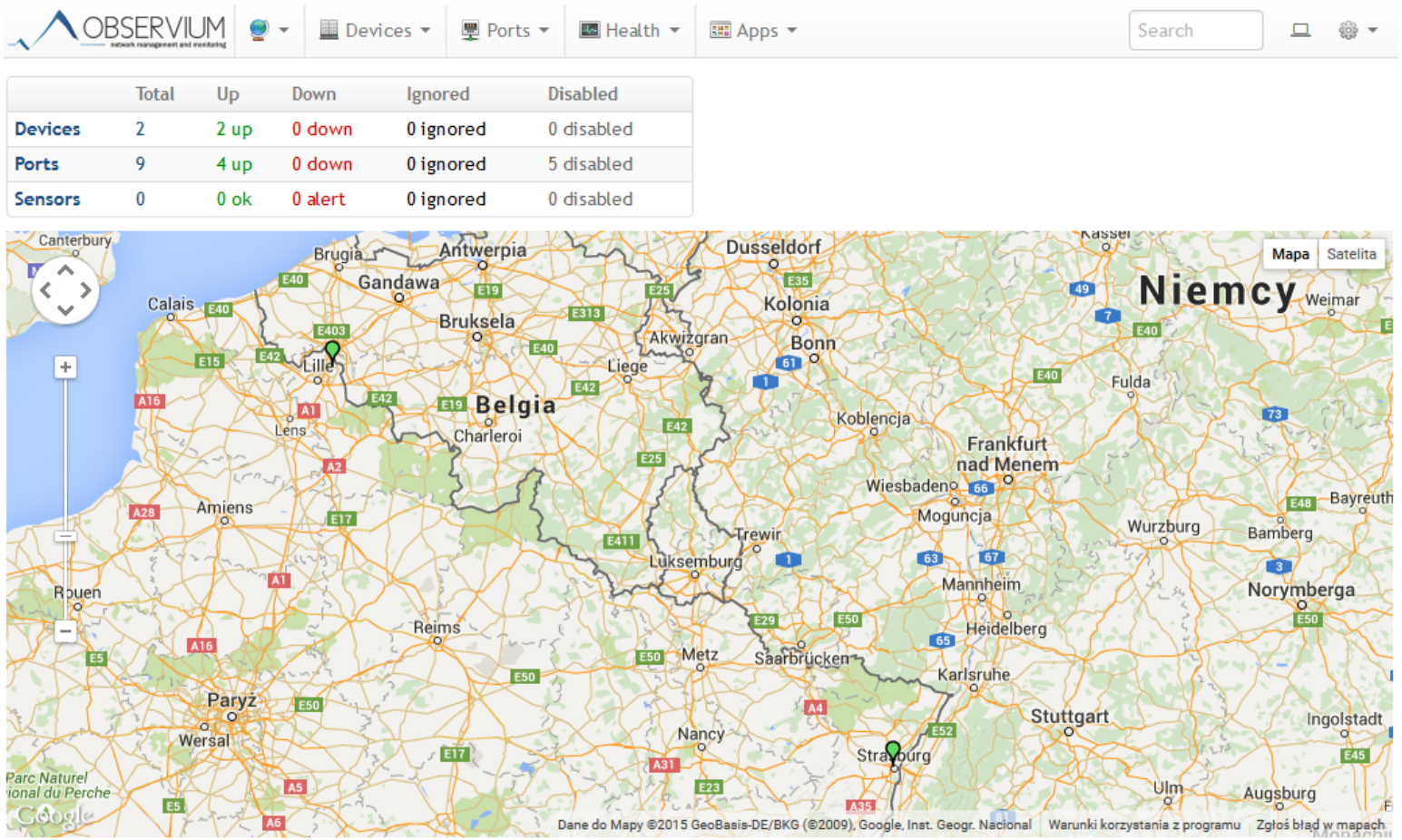
Założenia funkcjonalne systemu

- Automatyczne blokowanie adresów IP, z których dokonano ataków na serwer (na usługi serwera).
- Powiadamianie o błędach w konfiguracji usług na przykładzie usługi DNS.
- Monitorowanie bieżącego i historycznego obciążenia CPU.
- Monitorowanie wykorzystania CPU przez procesy PHP-FPM poszczególnych użytkowników i w przypadku przekraczania limitów zakańczanie pracy tych procesów.
- Monitorowanie pracy poszczególnych usług serwera takich jak Apache, PHP, Exim, Dovecot, MySQL i w przypadku wykrycia awarii usługi dokonania jej ponownego uruchomienia.
- Monitoring użycia twardego dysku lub macierzy dyskowej.

Założenia funkcjonalne systemu c.d.

- Wykonywanie automatycznych kopii bezpieczeństwa wszystkich monitorowanych serwerów.
- Kopie wykonywane na system plików umożliwiający tworzenie twardych dowiązań (twardych linków), dzięki którym te same pliki z różnych kopii jeśli nie różnią zajmują jedną i tą samą na dysku – oszczędność miejsca na dysku, na serwerze kopii bezpieczeństwa.
- Kompresja przesyłanych danych do kopii w trakcie połączenia, w celu oszczędzania ilości przesyłanych danych przez łącze internetowe (rsync). Wykluczenie danych zbędnych z kopii.
- Bezpieczeństwo przesyłanych danych (ssh).

Serwery VPS


OBSERVIUM
Devices
Ports
Health
Apps
Search
Total Up Down Ignored Disabled
Devices 2 2 up 0 down 0 ignored 0 disabled
Ports 9 4 up 0 down 0 ignored 5 disabled
Sensors 0 0 ok 0 alert 0 ignored 0 disabled
Mapa
Satelita
Niemcy
Belgia
Canterbury
Brugia
Antwerpia
Gandawa
Calais
Bruksela
Dusseldorf
Kolonia
Bonn
Akwizgran
Liege
Lille
Lens
Charleroi
Koblencja
Frankfurt nad Menem
Wiesbaden
Moguncja
Fulda
Weimar
Bayreuth
Wurzburg
Bamberg
Norymberga
Trewir
Luksemburg
Mannheim
Heidelberg
Karlsruhe
Stuttgart
Ingolstadt
Augsburg
Ulm
Amiens
Rouen
Reims
Metz
Saarbrücken
Nancy
Paryż
Wersal
Parc Naturel
Dane do Mapy ©2015 GeoBasis-DE/BKG (©2009), Google, Inst. Geogr. Nacional
Warunki korzystania z programu
Zgłoś błąd w mapach

Model systemu

Serwer monitorowany - Serwer monitorujący - Stanowisko administratora

Lista głównych skryptów

Lp.	Nazwa	Opis
1.	hardlink-backup-server.sh	Tworzenie kopii z twardymi linkami
2.	onekiller.sh	Pilnowanie limitów CPU użytkownika
3.	autofail2ban.sh	Generowanie czarnej listy adresów IP
4.	ipset.sh	Tablice hasz. dla czarnych list adresów IP
5.	rsync_firewall_serwer2.sh	Wysyłanie czarnych list na inne serwery
6.	firewall_ipv4.sh	Zapora sieciowa dla protokołu IPv4
7.	firewall_ipv6.sh	Zapora sieciowa dla protokołu IPv6
8.	named-checkconf.sh	Monitorowanie poprawności stref DNS
9.	pingtest-ipv4.sh	Test odpowiedzi na pingi - monitoring
10.	check_phpnodes.sh	Pilnowanie pracy serwerów PHP
11.	check-{service}.sh	Pilnowanie różnych usług serwera
12.	check_hdd_usage.sh	Monitoring użycia dysku
13.	check_all_snmp.sh	Monitoring innych zasobów (z SNMP)

Aplikacja QT C++ (GUI)

Monitoring wielu serwerów

Bezpieczeństwo

- Zastosowanie oprogramowania Rsync oraz SSH zapewnia wysoką niezawodność i bezpieczeństwo na etapie tworzenia połączenia przeznaczonego do przesyłania danych (kopie bezpieczeństwa).
- Wysyłanie powiadomień za pomocą e-maili zapewnia duże bezpieczeństwo i niezawodność przekazywania wiadomości do administratora systemu.

Wnioski

- Stworzony na potrzeby niniejszej pracy system, już funkcjonuje produkcyjne.
- Skrypty do tworzenia kopii bezpieczeństwa owych dwóch serwerów spisują się bardzo dobrze ponieważ na maszynie VPS wyposażonej w 100GB przestrzeń dyskową, znajduje kilkanaście pełnych kopii obu maszyn VPS, z których każda osobno posiada 25GB oraz 10GB danych, do zabezpieczenia.
- Skrypty wysyłające wiadomości e-mail z powiadomieniami o zdarzeniach sprawdzają się w praktyce. Wiele razy przyczyniły się do szybkiego rozwiązania awarii związanych z pojawiającym się błędem w konfiguracji usług lub z przeciążeniem serwera.
- Skrypt zapory sieciowej z powodzeniem zablokował w czasie trzech miesięcy pracy serwera około 10 tyś atakujących adresów IP, bez widocznego wzrostu obciążenia CPU.

Kierunki rozwoju

- Na pewno część skryptowa będzie bardzo intensywnie rozwijana wraz ze wzrostem ilości usług oraz użytkowników serwera. Będzie wymagana ich optymalizacja oraz zwiększenie ich możliwości jak w przypadku skryptu onekiller.sh, który będzie musiał zostać rozbudowany o nowe funkcje obsługujące procesy uprzywilejowane.
- Część aplikacyjna QT C++ będzie musiała zostać przygotowana na urządzenia mobilne. Koniecznością będzie poprawienie jej wyglądu i dostosowanie do urządzeń mobilnych.