Zabezpieczenie systemów i usług sieciowych Laboratorium 4

Zadanie 1 (*)

Celem zadania jest ustawienie automatycznej synchronizacji czasu z serwerem wzorcowym. Dzięki tej operacji możliwe jest właściwe skorelowanie zdarzeń z logów pomiędzy kilkoma serwerami. Do synchronizacji zazwyczaj stosowany jest protokół ntp. Nasz serwer zyskał automatyczną synchronizację czasu w momencie instalacji pakietu **openntpd**. Jednak program ten nie wykonuje skokowej synchronizacji czasu, minimalnie zwalnia on lub przyspiesza zegar komputera. W przypadku dużej rozbieżności zegara proces synchronizacji będzie długotrwały. Aby wykonać pierwsze skokowe nastawienie zegara instalujemy pakiet **ntpdate** i wydajemy polecenie: sudo ntpdate ntp.task.gda.pl. Aktualną datę i godzinę ustawioną w systemie możemy sprawdzić wydając polecenie date.

Zadanie 2 (*)

Celem zadania jest uruchomienie na serwerze systemu monitoringu. Posłużymy się narzędziem docker-compose (pakiet **docker-compose**). Po jego instalacji pobieramy pliki konfiguracyjne:

wget <u>https://zsius-pliki.justdoit.tech/docker-compose.yml</u> wget https://zsius-pliki.justdoit.tech/prometheus.yml

następnie wydajemy komendę:

docker-compose -f docker-compose.yml up -d

i dodajemy przekierowanie portu naszej maszyny wirtualnej 2230 -> 3000 (patrz lab1).

teraz możemy dostać się do grafany: <u>http://localhost:2230</u> (admin/sekret) i ustawić tam datasource jako typ prometheus ('Configuration' -> 'Data sources'), host <u>http://prometheus:9090</u>. Przykład:

Q		Ç Confi _{Organiza}	guration tion: Main Org.					
Q								
+		Data Sources	옷 Users	워. Teams 対	7 Plugins	¦li∤ Preferences	o [≁] API Keys	
88		Q Search by name						Add data source
Ø								
¢		Prome	theus default					PROMETHEUS
ø	Configuration	http://pror	netheus:9090					
Ū	Data Sources							
	옷 Users							
	ମ୍ଭ Teams							
	♥ Plugins							
	₩ Preferences							
	o [⊀] API Keys							

Po zapisaniu importujemy nowy dashboard (symbol '+' po lewej -> import) o id '1860':

+	Create	⊥ Upload JSON file	
88	器 Dashboard		
6	I Folder	Import via grafana.com	
^o	🛃 Import		
¢			Load
ŵ		Import via panel ison	
Ū			
		Load	

Zadanie 3 (*)

Celem zadania jest zapoznanie z podstawowym programem do wewnętrznego audytu systemu. Instalujemy pakiet **auditd**. Program audit po uruchomieniu rejestruje wszystkie istotne zdarzenia systemowe w pliku /var/log/audit/audit.log (może być skonfigurowany do wysyłania informacji na zdalny serwer). Przed uruchomieniem programu w pliku /etc/audit/auditd.conf ustawiamy opcję flush na SYNC oraz dopisujemy kilka reguł na końcu (zostawiamy obecną treść, dodajemy tylko nowe wpisy) pliku /etc/audit/rules.d/audit.rules:

```
-a exit,always -S unlink -S rmdir
-w /var/www -p wa
-w /etc/group -p wa
-w /etc/passwd -p wa
-w /etc/shadow -p wa
-w /etc/sudoers -p wa
```

Aby uruchomić program wydajemy komendę sudo systemctl restart auditd. Następnie aby zapisać jakieś logi instalujemy dowolny pakiet oprogramowania, np. ack-grep i wyświetlamy zawartość logu audytu poprzez sudo cat /var/log/audit/audit.log. Jakie informacje zawiera taki plik? Jak szybko przyrasta jego rozmiar?

Zadanie 4

Ponownie sprawdzić stronę monitoringu i sprawdzić obciążenie maszyny. Prawidłowo działające prometeusz i grafana powinny w tym momencie wyglądać mniej więcej tak:

Ø	器 Node Exporter Full ☆ ペ		🗤 🖧 🛞 📮 🕐 Last 1 hour 🗸							
0	datasource default ~ Job prometheus ~ Host: exporter:9100 ~									
α	~ Quick CPU / Mem / Disk									
+	் CPU Busy ⁱ sys Load (5m avg) ⁱ sys Load (15m av ⁱ RAM Used	i SWAP Used	i Root FS Used	ⁱ JPU Cor ~						
88				1	1					
Ø				ⁱ .tootFS	ⁱ ∢AM Total					
Д		N/A	N/A	N/A	2 GiB					
© •	~ Basic CPU / Mem / Net / Disk									
	i CPU Basic		i Memory Basic							
Q	100%	2.3 GiB								
	75%	1.9 GiB								
	50%	1.4 GiB								
	25%	954 MiB								
		477 MiB								
	^{U%} 22:30 22:40 22:50 23:00 23:10 23:20 — Busy System — Busy User — Busy Iowait — Busy IRQs — Busy Other — Idle	— 08 22:30 — RAM Total — RAM	22:30 22:40 22:50 23:00 23:10 — RAM Total — RAM Used — RAM Cache + Buffer — RAM Free — SWAP Used							
	i Network Traffic Basic		i Disk Space Used Basic							

Zadanie 5

Celem zadania jest zapoznanie się z podstawowymi narzędziami do analizy logów programu auditd. Narzędzia te to: ausearch oraz aureport. Pierwsze z nich pozwala na wygodne i szybkie przeszukiwanie często obszernych logów programu auditd. Uruchamiając je bez żadnych parametrów uzyskamy listę opcji dostępnych w programie.

Zadanie: Proszę wyszukać wszystkie rekordy zawierające w polu komendy (comm) słowo dpkg.

Narzędzie aureport pozwala na uzyskanie zbiorczego raportu na temat wszystkich zarejestrowanych zdarzeń. Jest to bardzo wygodne do automatycznej generacji zbiorczego raportu o stanie systemu.