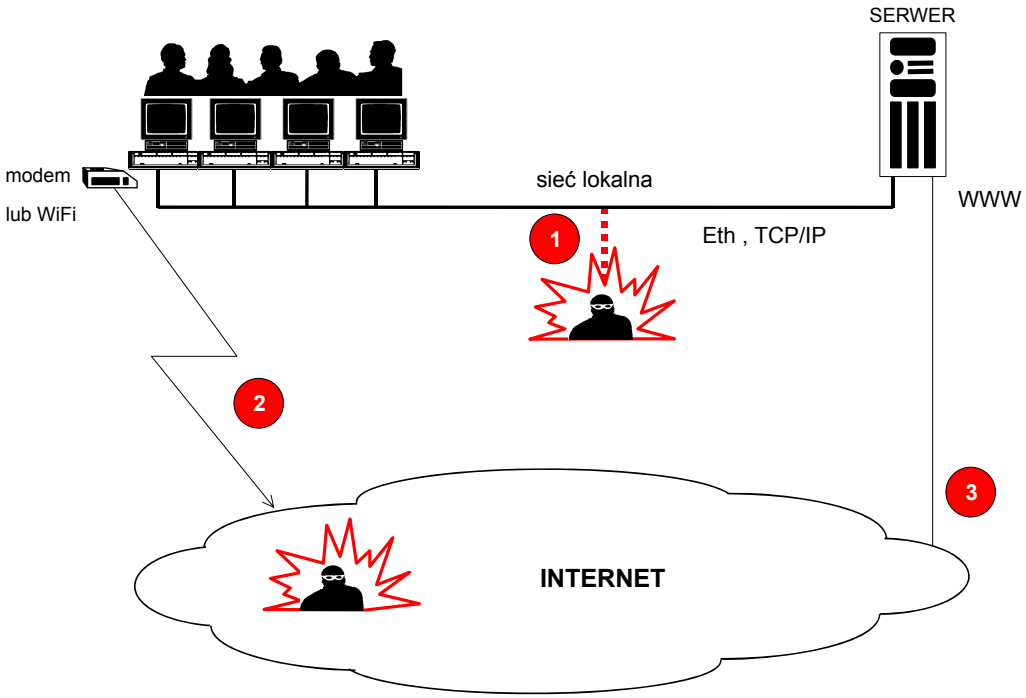
6. Podstawowe problemy bezpieczeństwa sieci komputerowych

Bieżący moduł przedstawia wybrane zagadnienia bezpieczeństwa związane bezpośrednio z niedoskonałościami współczesnych technologii sieci komputerowych. Omówione zostaną problemy bezpieczeństwa podstawowych protokołów sieciowych, klasyfikacja i przykłady ataków na środowiska sieciowe, podstawowe metody obrony oraz przykłady narzędzi podnoszących poziom bezpieczeństwa sieci

Rysunek 1 przedstawia schemat prostej sieci komputerowej analizowany w dalszej części modułu jako przykładowy scenariusz ataków na środowisko sieciowe i obrony przed tymi atakami. Rozważać będziemy sieć lokalną hipotetycznego przedsiębiorstwa lub instytucji, obejmującą serwer aplikacyjny, np. z systemem bazy danych zawierającej dane przetwarzane w firmie, zbiór stacji roboczych, na których działają klienckie aplikacje użytkowników wykorzystujące informacje z bazy danych. Oprócz infrastruktury sieci lokalnej wyróżnimy jeszcze łącze do sieci rozległej. Przyjmijmy dla uatrakcyjnienia rozważań, iż jest to sieć publiczna, np. Internet. Utrzymywanie łącza do sieci rozległej może być wymagane chociażby z tego powodu, że pewne ograniczone informacje z bazy danych firma chce publicznie udostępniać swoim klientom poprzez usługę WWW. Oczywiście możliwych jest kilka punktów dostępowych do sieci rozległej, mogą to być np. łącza modemowe, analogowe lub cyfrowe typu xDSL, lub sieci bezprzewodowe z dostępem do Internetu.



Rysunek 1. Schemat sieci komputerowej analizowany jako scenariusz zagrożeń

Rozważany scenariusz zagrożeń bezpieczeństwa obejmuje następujące przypadki:

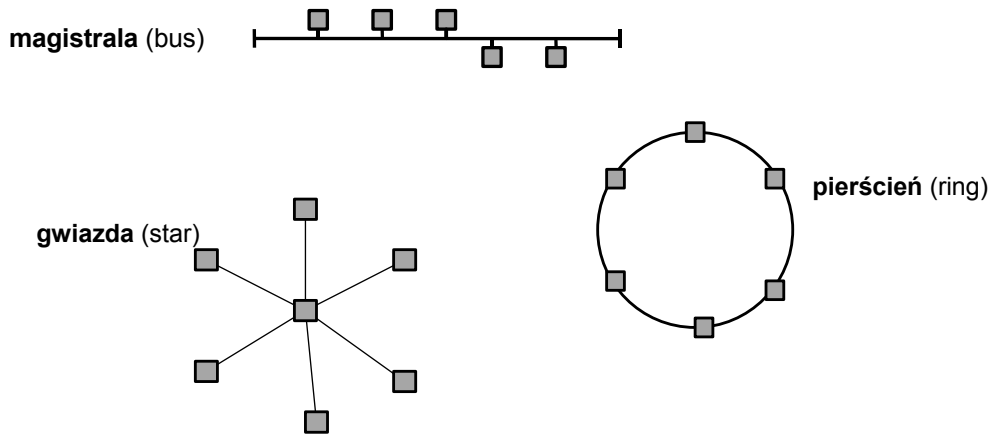
1. Uzyskanie dostępu do konta w systemie / bazie danych, z czym wiąże się możliwość:
    - naruszenia własności poufności / integralności / dostępności przechowywanych w systemie danych
    - rekonfiguracji systemu (co w istocie jest też naruszeniem własności integralności)
2. Pozyskanie / modyfikacja transmitowanych danych
    - naruszenia własności poufności / integralności / dostępności danych transmitowanych między serwerem a stacjami roboczymi
3. Rekonfiguracja sieci (urządzeń sieciowych, protokołów) – naruszenie integralności
4. Zablokowanie funkcjonowania stacji / urządzeń sieciowych i w efekcie naruszenie własności dostępności informacji

Przyczyn tych zagrożeń należy przede wszystkim szukać w niedoskonałościach technologii sieciowych wykorzystywanych aktualnie. Należy od razu zaznaczyć, niejako tytułem usprawiedliwienia, iż większość tych technologii zostało zaprojektowanych wiele lat temu (pierwotne wersje niektórych rozważanych standardów datują się na wczesne lata 70-te ubiegłego wieku) i do dziś były zaledwie modernizowane na ogół w celu usprawnienia działania, dostosowania do nowych wymagań, a raczej rzadko dla nieznacznego podniesienia bezpieczeństwa.

Obserwując właściwości technologii należących do kolejnych warstw modelu referencyjnego OSI, interesujących odkryć możemy dokonać już w warstwie pierwszej – fizycznej. Poniżej przypomniane zostaną, raczej jedynie hasłowo, tytułem zaakcentowania, najistotniejsze z tych właściwości.

Elementami specyfikacji funkcjonalnej tej warstwy są topologie fizyczne i media komunikacyjne.

Ze względu na własności zarówno poufności, integralności jak i dostępności możemy łatwo uszeregować typowe topologie sieciowe o najmniej do najbardziej bezpiecznej (zastanów się jak wygląda to uszeregowanie). Szczęśliwie najkorzystniej wypada tu najpowszechniej dziś spotykana topologia gwiazdy (przypomnij sobie jej własności).



Rysunek 2. Modele typowych topologii sieciowych

Również typowo dziś spotykane media można podobnie uszeregować, od najmniej bezpiecznych z natury – mediów bezprzewodowych, poprzez różne technologie skrętki komputerowej, niechaj wymienić tu: UTP, FTP, STP czy SSTP, aż po światłowód.

W warstwie łącza danych pojawiają się już elementy inteligencji sieciowej – protokoły komunikacyjne. Najistotniejszym współcześnie jest niewątpliwie protokół Ethernet w różnych stosowanych standardach.

Z punktu widzenia bezpieczeństwa architekturę sieci Ethernet wyróżniają następujące cechy:

- ruch niejawnie rozgłoszeniowy, współdzielenie medium (topologia logiczna)
- komunikacja jawnie rozgłoszeniowa (adresy rozgłoszeniowe i grupowe)
- możliwość pracy sterownika sieciowego w trybie diagnostycznym (ang. *promiscuous*)
- stosowane proste liniowe sumy kontrolne (CRC) dla kontroli integralności transmitowanych ramek

Funkcjonujące w tej warstwie urządzenia sieciowe (mosty i przełączniki) charakteryzują następujące cechy:

- mostowanie transparentne (TB) i źródłowe (SR) i związana z tymi funkcjami selektywna propagacja danych (prosta lecz podatna na nadużycia odmiana filtracji)
- automatyka obsługi dowolnych stacji i współpracy z innymi urządzeniami (np. protokół drzewa rozpinającego STP)

Z racji szczególnej roli wszechobecnej dziś rodziny protokołów TCP/IP, spośród kolejnych warstw modelu OSI najistotniejsze są te, które odwzorowują się najdokładniej na model internetowy. Dotyczy to warstwy sieciowej, transportowej oraz aplikacyjnej.

W warstwie sieciowej szczególną uwagę należy zwrócić na problematykę:

- bezpołączeniowa semantyki i zawodnej transmisji pakietów (datagramów) IP
- mechanizmów zautomatyzowanego wsparcia dla adresacji: ARP, RARP
- funkcjonalności routingu dynamicznego
- kapsułkowania (kopertowania) pakietów.

Zagadnienia dotyczące warstwy sieciowej są na tyle ważne, iż wymienione wyżej problemy zostaną omówione szerzeń w dalszej części bieżącego modułu. Wcześniej jednak podkreślone zostaną istotne cechy pozostałych ważnych warstw.

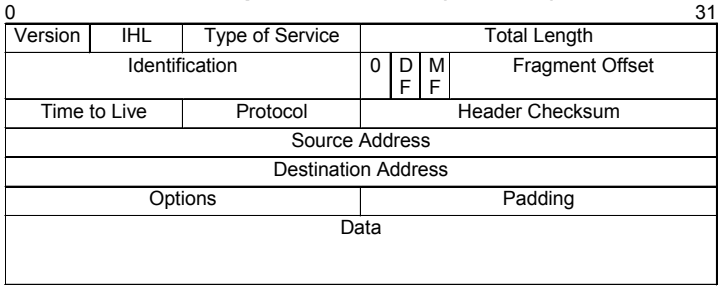
Warstwa transportowa cechuje się funkcjonalnością której najistotniejsze elementy określa się mianem sterowania przepływem.

Natomiast w warstwie aplikacyjnej pojawiają się problemy związane z niedoskonałościami popularnych protokołów (telnet, ftp, SMTP, POP, IMAP) na ogół nie dysponujących mechanizmami ochrony poufności oraz integralności i stosującymi mało wiarygodne mechanizmy uwierzytelniania.

Warstwa sieciowa

Protokół IP, jest najpowszechniej spotykanym protokołem sieciowym i analizę problemów bezpieczeństwa ograniczamy wyłącznie do aspektów dotyczących tego protokołu.

PDU protokołu IP v.4 (RFC-791)



Rysunek 3. Format datagramu IP

Adresacja

Adresacja jest jednym z zadań podstawowych protokołów warstwy sieciowej. Pola adresowe ustawiane są w nagłówku datagramu IP przez stację nadawczą (źródło, ang. *source*). Z zawartością pól adresowych wiążą się następujące problemy:

- nie ma gwarancji, że pakiet został wysłany z adresu wpisanego w polu *Source Address*
- wiele systemów kontroluje to pole w momencie wysyłania datagramu
- niektórzy operatorzy dostępu do Internetu stosują filtry blokujące pakiety z nieprawidłowym adresem źródłowym
- mimo tego nie można polegać na poprawności adresu źródłowego odebranego pakietu.

Problemy te dyskwalifikują uwierzytelnianie poprzez wartość pola adresu źródłowego. Niestety wiele protokołów aplikacyjnych posiada wbudowane takie mechanizmy. Ewentualny atak na system może zatem polegać sfałszowaniu adresu źródłowego (*IP spoofing*). Kompromituje to procedury uwierzytelniania ofiary.

Trasowanie

Trasowanie (routing) polegające na wyznaczaniu drogi do adresata. W przypadku protokołu IP trasowanie jest wykonywane pojedynczymi etapami. Zadanie to wykonuje router. Trasowanie jest niezwykle istotnym elementem działania warstwy sieciowej i związanych z nim jest kilka problemów:

- przeciążony router może odrzucać nadchodzące pakiety
- za retransmisję odpowiadać muszą protokoły warstw wyższych

- jeśli router zostanie „zalany” bardzo dużą masą pakietów (nieistotne czy prawidłowych), to ewentualne przeciążenie doprowadzi do zablokowania komunikacji (pakietów należących do aktywnych sesji – asocjacji IP)

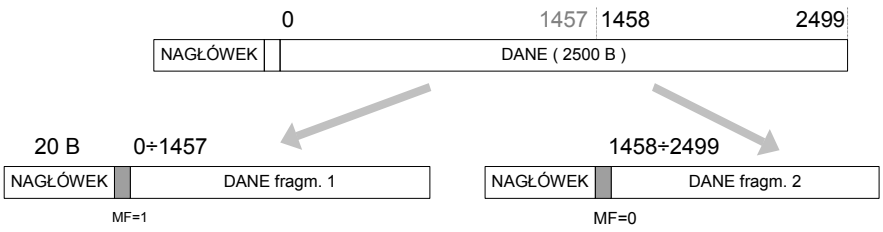
Przeciążenie routera implikuje zagrożenia dostępności danych transmitowanych w pakietach kierowanych do niego. Co więcej, potencjalny atak zdalny skierowany przeciwko innym stacjom sieciowym może wykorzystywać chwilową niedostępność pakietów z oryginalnego źródła celem podszycia się pod źródłowy komputer.

Fragmentacja pakietów

Fragmentacja datagramów IP jest często realizowaną funkcją, pierwotnie zamierzoną jako mechanizm optymalizacji kosztu przetwarzania pakietów. O wielkości fragmentów decyduje wartość MTU (*Maximum Transfer Unit*) związana z wielkością pola danych ramek warstwy drugiej OSI. Do fragmentacji dochodzić może na dowolnym etapie drogi między nadawcą a odbiorcą. Oryginalny datagram otrzymuje unikalny (w przybliżeniu) identyfikator, którym opatrywane są kolejne fragmenty. Kolejność fragmentów determinuje względny numer pierwszego bajtu przekazywanego w polu danych fragmentu liczony od początkowego bajtu oryginalnego datagramu (*Fragment Offset*). Scalanie fragmentów odbywa się na węźle odbiorcy.

Fragmentacja może stanowić potencjalne źródło zagrożenia a następujących powodów:

- każdy fragment jest również datagramem i jako taki może mieć, teoretycznie, do 64 kB wielkości – z premedytacją spreparowane fragmenty przekraczające w sumie 64 kB mogą powodować błędy scalania
- podobnie manipulacje wartościami pola *Fragment Offset*
- niektóre stacje IP, w tym zabezpieczenia (filtry pakietów), mogą zachowywać się niepoprawnie atakowane przez niewłaściwie pofragmentowane pakiety – zjawisko to wykorzystuje np. *teardrop attack* [Ziemba, RFC 1858]
- często filtry przetwarzają właściwie (np. odrzucają) tylko pierwszy fragment pakietu – wynika to z faktu iż w praktyce informacje niezbędne do przeprowadzenia filtracji znajdują się tylko w pierwszym pakiecie, a ponadto jest to działanie wystarczające do wykluczenia poprawnego scalenia niepożądanego datagramu w węźle odbiorcy, jednak powoduje to mimowolne przepuszczanie kolejnych fragmentów należących do (będących przecież kontynuacją) ruchu sklasyfikowanego jako niepożądany. Fakt przepuszczania takich pakietów może zostać wykorzystany do realizacji niektórych typów ataków.



Rysunek 4. Schemat fragmentacji IP

Pakiety rozgłoszeniowe

Mechanizmy rozgłoszeniowe oferuje wiele protokołów, również IP. Ukierunkowane rozgłoszenie często jest wykorzystywane do przeprowadzenia ataków na dostępność informacji (typu *Denial of Service – DoS*). Stanowi to najistotniejsze zagrożenie związane z mechanizmem rozgłaszania. Szczęśliwie, wiele routerów posiada funkcję blokowania ruchu rozgłoszeniowego.

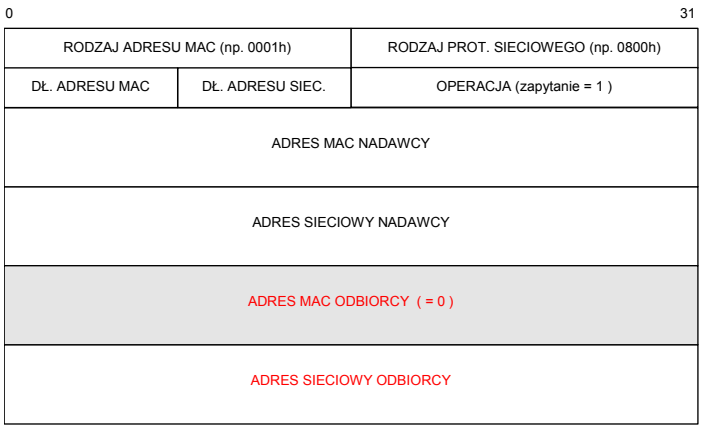
Odwzorowanie adresów

Odwzorowanie adresów IP na adresy MAC (np. Ethernet) jest niezbędne dla realizacji operacji nadawczych, a dokładniej do konstrukcji prawidłowej ramki MAC. Zadaniem odwzorowania adresów na ogół zajmuje się protokół ARP (*Address Resolution Protocol*). Stosuje on transmisję rozgłoszeniową zapytań i zbiera odpowiedzi bez zapewnienia poufności i autentyczności. W celu poprawy efektywności, protokół wykorzystuje pamięć podręczną do temporalnego składowania informacji pozyskanych z docierających zapytań i odpowiedzi ARP.

W efekcie z protokołem ARP wiążą się następujące zagrożenia:

- stacja w sieci lokalnej może wysyłać fałszywe zapytania lub odpowiedzi ARP
- kierując w efekcie inne pakiety w swoim kierunku (*ARP spoofing*)
- dzięki czemu napastnik może:
  - modyfikować strumienie danych
  - podszywać się pod wybrane komputery

Często można skonfigurować lokalne statyczne mapowanie ARP wyłączając przy tym automatyczną obsługę zapytań i odpowiedzi ARP, co uwalnia od w/w problemów. Jednak w praktyce okazuje się, że wyłączenie obsługi komunikacji sieciowej nie jest możliwe w każdej implementacji stacji protokołu ARP.

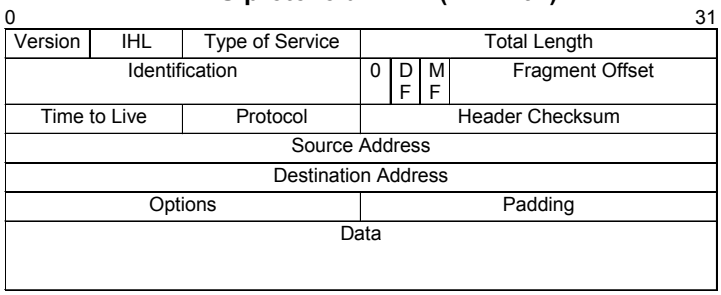


Rysunek 5. Format zapytania ARP

Warstwa transportowa

Jak wiadomo, w rodzinie protokołów internetowych występują 2 protokoły transportowe. Protokół TCP (*Transmission Control Protocol*) jest to protokół strumieniowy zorientowany połączeniowo. Zestawienie komunikacji w protokole TCP wymaga wykonania 3-etapowej procedury nawiązania połączenia (*3-way handshake*). Bodaj najistotniejszym zadaniem tej procedury jest ustalenie inicjalnych numerów sekwencyjnych, rozpoczynających numerowanie bajtów strumienia danych przekazywanego w każdym z dwu kierunków połączenia A-B (rysunek 7).

PDU protokołu IP v.4 (RFC-791)



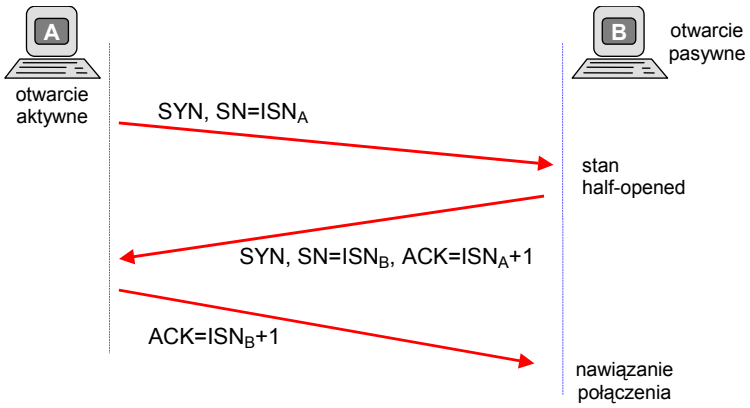
Rysunek 6. Format segmentu TCP

Na schemacie z rysunku 7 zastosowano następujące oznaczenia:

SN = Numer sekwencyjny w nagłówku segmentu określa numer pierwszego oktetu danych przesyłanych w tym segmencie

ACK = Numer potwierdzenia – numer sekwencyjny następnego oktetu danych po ostatnim pomyślnie odebranym (numer oczekiwanego oktetu)

ISN = Inicjalny numer sekwencyjny (*Initial Sequence Number*) – początkowy numer sekwencyjny danych przesyłanych w danym połączeniu, ustalany w procesie nawiązania połączenia (w segmencie SYN). Każde połączenie może rozpocząć numerację oktetów danych od arbitralnej wartości (jeśli ISN=0, to pierwszy oktet w całym połączeniu ma numer ISN+1=1). Inicjalny numer sekwencyjny jest ustalany oddzielnie dla obu stron połączenia (w ogólności $ISN_A \neq ISN_B$)



Rysunek 7. Schemat nawiązania połączenia (3-way handshake)

Nawiązanie połączenia TCP charakteryzują następujące cechy:

- możliwe jest zdalne wymuszenie połączenia i przechwycenie legalnej komunikacji nawet bez odbioru segmentu SYN+ACK
- wymaga to odgadnięcia ISN zawartego w tym segmencie
- na szczęście numery ISN są wybierane pseudolosowo
- na ogół z rozkładem bardzo dalekim od losowego
  - wg sugestii z RFC 793 – licznik ISN jest inkrementowany co 4 μs
  - starsze jądra wywodzące się z BSD (4.2) dokonują inkrementacji o wartość stałą co 1 sek i przy każdym nowym połączeniu
- więc łatwo przewidzieć ISN dla nowych połączeń

Stopień randomizacji wyboru ISN graficznie reprezentuje się wykresem fazowym. Wykresy fazowe prawidłowego generatora ISN zgodny ze standardem RFC 1948 oraz generatorów w popularnych (mniej i bardziej) systemach operacyjnych można znaleźć w <A HREF="http://alon.wox.org/tcpseq/">[Zalewski]</A>. Jak pokazują przykłady wykresów fazowych, coraz więcej systemów operacyjnych stosuje generatory zbliżone do wzorcowego, jednak nadal wiele jest implementacji dalece od wzorca odbiegających. Skutkuje to możliwością przeprowadzenia pewnych ataków, o których będzie mowa dalej.

Procedura nawiązania połączenia TCP może być wykorzystana do realizacji następujących ataków:

- ataki zalewające ofiarę nowonawiązywanymi połączeniami – dla każdego nawiązanego połączenia system przydziela pewne zasoby, w szczególności pamięć. Zasoby zwalniane są po zamknięciu połączenia. Jeśli zamknięcie nie następuje, wówczas zasoby pozostają zajęte (mimo iż mogły w ogóle nie być wykorzystane). Odpowiednio duża liczba zestawionych połączeń może doprowadzić do przydzielenia im całej dostępnej pamięci nie pozostawiając żadnych dostępnych zasobów do pracy systemu i powodując załamanie jego pracy.
- ataki zalewające ofiarę segmentami SYN (wykorzystujące stan *half-opened*) – w praktyce system operacyjny przydziela zasoby dla nowozestawianego połączenia,

jak tylko pojawia się pierwszy segment SYN, jeszcze zanim 3-etapowa procedura nawiązywania zostanie zakończona. Zasoby zajmowane są zatem nawet jeśli połączenie nie zostanie w pełni zestawione. Duża liczba na wpół nawiązanych połączeń może wyczerpać zasoby i ostatecznie blokuje stację protokołu TCP i cały system. Co więcej, takie ataki są trudno wykrywalne, bowiem typowe implementacje TCP nie raportują wyższym warstwom OSI (systemowi operacyjnemu) żadnych zdarzeń związanych z połączeniem, które nie jest jeszcze w pełni nawiązane.

Warstwa aplikacyjna

W warstwie aplikacyjnej występują m.in. problemy trywialnego uwierzytelniana na podstawie identyfikacji usług oraz braku odpowiednich zabezpieczeń usług narzędziowych.

Identyfikacja usług

Identyfikacja usług w modelu internetowym odbywa się zwykle jedynie na podstawie numer portu źródłowego lub docelowego. Z wykorzystaniem wartości numeru portu wiążą się następujące obserwacje:

- lokalny numer portu klienta jest niemal zawsze wybierany przypadkowo przez system operacyjny (choć klient może go wybrać sam)
- w systemie Unix występują tzw. porty uprzywilejowane (systemowe) – są to porty o numerach do wartości 1024. Pod tymi portami system pozwala uruchomić proces sieciowy tylko administratorowi (root). Teoretycznie zatem, można domniemywać iż uruchomione pod systemowymi portami procesy należą do zaufanych i bezpiecznych. Jednak w praktyce, nie ma oczywiście pewnego sposobu zweryfikowania prawdziwości tego domniemania zdalnie.
- niestety nadal w wielu przypadkach systemy zdalne polegają na zaufaniu do asocjacji obejmujących te porty (połączeń z tymi portami)
- w istocie restrykcja wykorzystania portów uprzywilejowanych tylko przez administratora jest wyłącznie konwencją – nie należy do specyfikacji protokołów usług, a co więcej – nie dotyczy systemów innych niż Unix
- poleganie na niej absolutnie nie jest bezpieczne

Usługi narzędziowe infrastruktury sieciowej

Jednym z najpopularniejszych przykładów usług narzędziowych jest DNS (*Domain Name Service*). Jak wiadomo, system DNS to swoista rozproszona baza danych odwzorowań nazwa domenowa ↔ adres sieciowy. Baza DNS ma strukturę drzewiastą, poddrzewa odpowiadają poszczególnym domenom niższego poziomu (poddomenom). Zarządzanie poddrzewami może być delegowane innym serwerom DNS. Aktualizacje bazy DNS mogą obejmować pojedyncze rekordy RR (*resource records*), jak i całe poddrzewa. Poprzez protokół DNS można dokonywać prostych zapytań o pojedyncze odwzorowania, jak i zrealizować pozyskanie pełnej kopii fragmentu obszaru nazw (tzw. transfer stref – *zone transfer*) np. w celu aktualizacji serwerów zapasowych. Protokół DNS dostępny jest poprzez oba internetowe protokoły transportowe: UDP – tak realizowane są proste zapytania DNS, TCP – tak odbywa się transfer stref DNS.

Z punktu widzenia bezpieczeństwa istotne jest, iż niektóre zapisy RR dostarczają informacji użytecznych dla włamywaczy, np. HINFO (może zawierać m.in. informacje o systemie operacyjnym), WKS (well-known-services). Pola te, na szczęście, rzadko są dziś stosowane.

Baza DNS składa się z dwóch oddzielnych drzew mapowań – dla mapowania nazw na adresy (zapytania proste) i adresów na nazwy (zapytania odwrotne – *inverse queries*). Nie ma wymuszonej relacji między drzewami – każde z nich jest w praktyce utrzymywanie niezależnie. Przy tym drzewo mapowań odwrotnych zwykle nie jest równie często aktualizowane a do tego w ogóle jest utrzymywane w gorszym stanie. Niestety stanowi to potencjalne ułatwienie w przejęciu kontroli nad częściami drzewa mapowań odwrotnych i, w efekcie, skutecznym podszywaniu się pod autoryzowane nazwy.

W przypadku usługi DNS można wyróżnić następujące podstawowe problemy dotyczące bezpieczeństwa:

- udostępnianie użytecznych informacji atakującym
- brak uwierzytelniania w protokole zapytań DNS i transferu stref, co umożliwia fałszowanie danych (*pharming*)
- podszywanie się pod autoryzowne nazwy kompromituje system uwierzytelniania przez nazwę – w tym przypadku możliwa jest jednak prosta obrona przez dodatkową weryfikację w drzewie mapowań nazw i wykrycie fałszerstwa
- możliwość „zatruwania" fałszywymi odpowiedziami lokalnej pamięci *cache* usługi DNS stacji uwierzytelniającej jeszcze zanim wyśle ona zapytanie o mapowanie – wówczas fałszerstwo nie wyjdzie na jaw

Usługi inne popularne usługi narzędziowe BOOTP i DHCP również udostępniają informacje o infrastrukturze sieciowej, i to często bardzo bogate informacje, praktycznie bez uwierzytelniania. Na szczęście dostępne są one na ogół tylko w obrębie sieci lokalnej, zatem mogą być wykorzystane tylko przez atakujących, którzy wdarli się już do atakowanej podsieci. Istotny jest jednak problem bezpiecznej wymiany danych pomiędzy serwerami DHCP a DNS, a ta z kolei może przechodzić przez kilka podsieci.

Typowe ataki na infrastrukturę sieciową

Powszechne techniki ataków na infrastrukturę sieciową wykorzystują głównie niedoskonałości protokołów oraz technologii sieciowych w celu:

- uzyskania danych (*information recovery*)
- podszycia się pod inne systemy w sieci (*host impersonation*)
- manipulacji mechanizmami dostarczania pakietów (*temper with delivery mechanisms*)

Poniżej wymienione zostaną najczęściej spotykane współcześnie techniki ataków zebrane w następujące 4 klasy:

1) Sniffing/scanning:

- *network sniffing* – jest to pasywny podgląd medium transmisyjnego, np. w celu przechwycenia interesujących ramek (*packet snooping*)

- *network scanning* – jest to wykorzystanie specyfiki implementacji protokołów do sondowania (*enumeration*) urządzeń aktywnych w sieci, aktywnych usług, konkretnych wersji systemu operacyjnego i poszczególnych aplikacji (sztandarowym przykładem narzędzi realizujących taki atak jest program *nmap*)

2) Spoofing:

- *session hijacking* – przejmowanie połączeń poprzez „wstrzelenie" odpowiednio dobranych pakietów – wymaga dostępu do uprzednio legalnie zestawionego połączenia TCP
- *TCP spoofing* – podszywanie bazujące na oszukaniu mechanizmu generowania numerów ISN; wykorzystanie ataku np. w celu oszukania mechanizmów uwierzytelniania usług r* (które dokonują uwierzytelniania przy użyciu funkcji rusersok())
- *UDP spoofing* – prostsze od TCP w realizacji (ze względu na brak mechanizmu szeregowania i potwierdzeń ramek w protokole UDP), użyteczne podczas atakowania usług i protokołów bazujących na UDP np. DNS.

3) Poisoning:

- *ARP spoofing/poisoning* – wykorzystuje zasady działania protokołu ARP, umożliwiając zdalną modyfikację wpisów w tablicach ARP systemów operacyjnych oraz przełączników, a przez to przepełnianie tablic ARP
- *DNS cache poisoning* (*pharming*, także znany jako *birthday attack*) – umożliwia modyfikację wpisów domen w dynamicznym *cache* DNS, co jest niezwykle dużym zagrożeniem w połączeniu z atakami pasywnymi
- *ICMP redirect* – wykorzystanie funkcji *ICMP* do zmiany trasy routingu dla wybranych adresów sieciowych
- ataki na urządzenia sieciowe przy pomocy protokołu SNMP

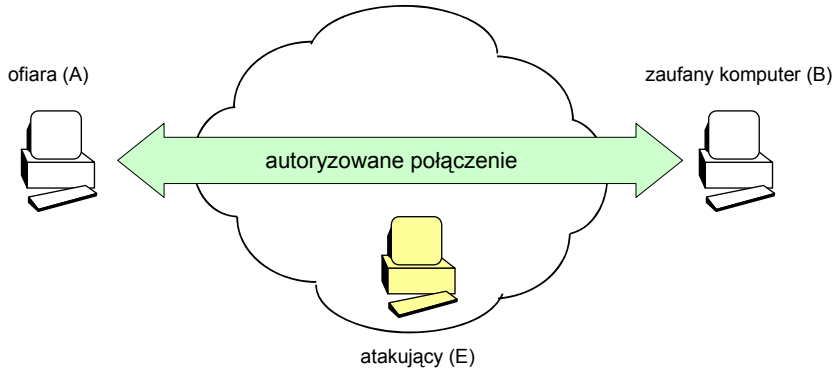
4) Denial of Service (DoS)
w tej kategorii mieszczą się wszystkie te ataki, których ostatecznym celem jest unieruchomienie poszczególnych usług, całego systemu lub całej sieci komputerowej.

- oto przykłady kilku najpopularniejszych ataków:
  - SYN flood
  - smurf, fraglle
  - land
  - tribal flood (trin00, trinio, trinity v3)
  - subseven, stacheldracht
  - UDP storms (teardrop, bonk)
  - ICMP destination unreachable
- istnieją też ataki na wyższe warstwy modelu OSI, np. e-mail bombing.

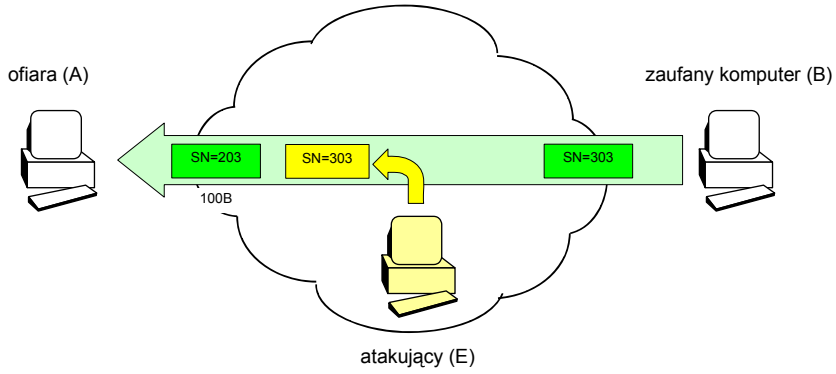
Atak session hijacking

Rysunek 8 przedstawia wyjściowy stan ataku *session hijacking*, którego celem jest nieuprawnione wpięcie segmentów protokołu transportowego w strumień wymieniany w autoryzowanym (poprawnie zestawionym) połączeniu między systemem A (przyszłą ofiarą

ataku) i zaufanym systemem B. Atakujący E, mając wgląd w dotychczasową zawartość strumienia w kierunku do B do A (poprzez *sniffing*), może spreparować poprawny i oczekiwany przez A segment, który podsunie jako rzekomo autentyczny segment wysłany przez B (rysunek 9).



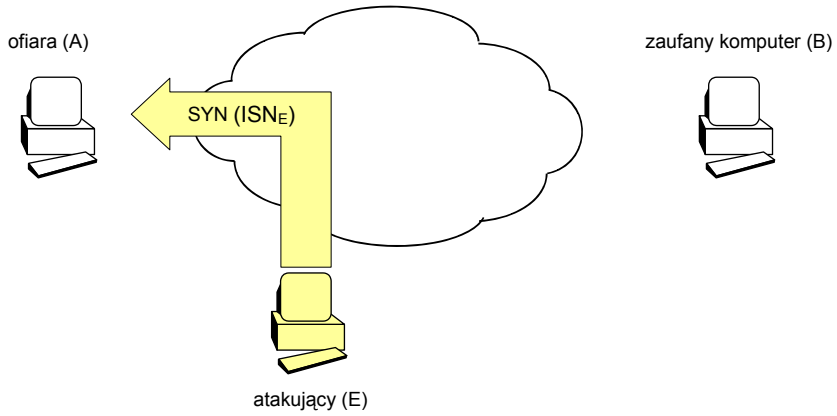
Rysunek 8. Schemat ataku session hijacking (1)



Rysunek 9. Schemat ataku session hijacking (2) – w strumieniu przesyłane są segmenty po 100 B

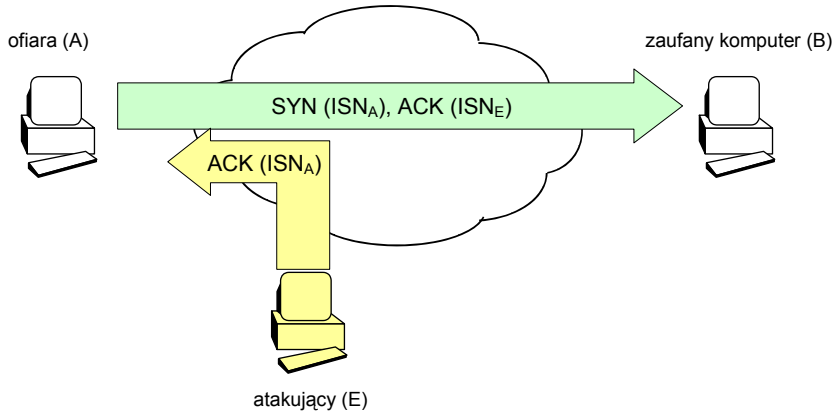
Atak TCP spoofing

Rysunek 10 przedstawia początek ataku *TCP spoofing*, którego celem jest nieuprawnione zestawienie połączenia z systemem A (ofiarą ataku) w imieniu zaufanego systemu B. Atakujący E tym razem nie ma wglądu komunikację między A i B, co czyni atak znacznie trudniejszym niż *session hijacking*. Atak wymaga nie tylko sfałszowania adresu źródłowego (w szczególności w pierwszym segmencie SYN, nawiązującym połączenie), ale dodatkowo poprawnego przewidzenia numeru ISN_A (który zaproponuje system A w drugim segmencie SYN/ACK) i być może jeszcze zablokowania poprawnej komunikacji z rzeczywistym systemem B (co może wymagać przeprowadzenia ataku DoS skierowanego przeciw B), aby B nie mógł zakończyć (zresetować) niechcianego połączenia. Mimo tych utrudnień E może spreparować poprawny i oczekiwany przez A segment ACK – kończący poprawnie procedurę nawiązania połączenia, (rysunek 11).



Rysunek 10. Schemat ataku TCP spoofing (1)

Najtrudniejszym krokiem ataku jest wysłanie poprawnego segmentu trzeciego zawierającego potwierdzenie inicjalnego numeru sekwencyjnego ISN_A wybranego przez A. Ze względu na brak możliwości podglądu przez E komunikacji z A do prawdziwego B, wymaga to przewidzenia wartości ISN_A przez E. Jest to prawdopodobne przy wygenerowaniu relatywnie niedużej liczby segmentów ACK, jeśli system A nie posiada poprawnego generatora ISN. Wówczas jest szansa, iż jeden z wygenerowanych przez E segmentów będzie przenosił poprawną wartość potwierdzenia i zostanie przez A uznany za oczekiwany.



Rysunek 11. Schemat ataku TCP spoofing (2)

Można wyróżnić następujące elementy ataku TCP spoofing:

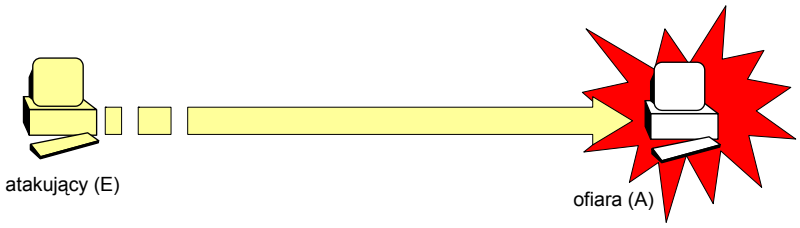
- wytypowanie właściwego ISN_A – np. poprzez uprzednie nawiązanie autoryzowanego połączenia na inny port (pozyskanie wcześniejszego numeru ISN'_A)
- celowe uniemożliwienie przetwarzania segmentu drugiego, zawierającego wartości SYN (ISN_A) i ACK (ISN_E)

Poczynić można w związku z tym następujące obserwacje:

- atakujący nie musi mieć dostępu do segmentów autoryzowanego połączenia
- jeśli ma (*sniffing*) – może podejrzeć numer sekwencyjny (nie musi zgadywać)
- i podpiąć się na dowolnym etapie już zestawionego połączenia

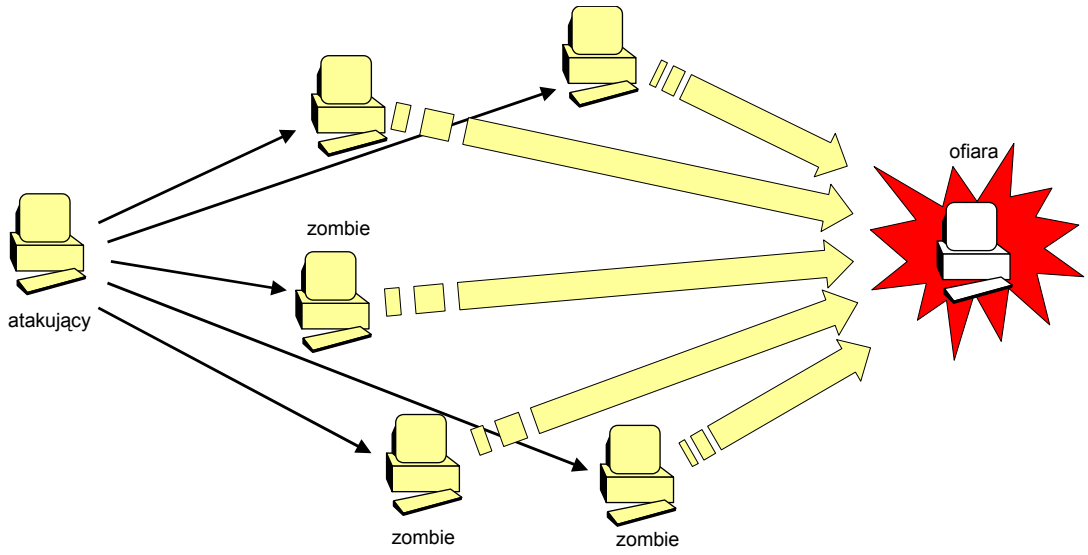
Atak Denial of Service (DoS)

Celem ataku DoS jest unieruchomienie całego systemu ofiary lub jego komponentów (rysunek 12). W tym celu stosowane są różne techniki ataku. Niektóre z nich zostaną krótko przedstawione poniżej.



Rysunek 12. Schemat ataku Denial of Service (DoS)

Szczególnie niebezpieczną odmianą ataku jest rozproszony DoS (Distributed Denial of Service – DdoS), w którym atakujący nie przeprowadza ataku bezpośrednio, lecz doprowadza do skomasowanego natarcia wykorzystując inne systemy (często w dużej liczbie). Zwykle owe systemy uczestniczące mimowolnie w skomasowanym ataku, zostały wcześniej opanowane przez atakującego i tak odpowiednio zmodyfikowane by ułatwić mu w przyszłości przeprowadzanie ataku DDoS.



Rysunek 13. Schemat ataku Distributed DoS (DDoS)

Atak SYN flood

W przypadku ataku SYN flood atakujący (E) wysyła na adres ofiary (A) dużą liczbę segmentów SYN protokołu TCP adresowanych z dowolnych (nieistniejących) adresów IP. Nieświadomy tego A odpowiada segmentami SYN/ACK i rozpoczyna bezowocne oczekiwanie na segmenty ACK (stacja protokołu TCP ofiary jest w stanie na wpół otwartym). W trakcie oczekiwania wyczerpują się zasoby stacji protokołu TCP i systemu operacyjnego A.

W 1997 r. atak SYN flood na WebCom wyłączył z użycia ponad 3000 witryn WWW

Ping of Death

Atak ten przeprowadza się poprzez wygenerowanie pofragmentowanych pakietów ICMP przekraczających w sumie 64kB. Wówczas scalanie może w niektórych implementacjach powodować błędy prowadzące do zawieszenia stacji IP.

Smurf attack

Jest to atak DDoS. Polega na wygenerowaniu dużej ilości rozgłoszeniowych (*directed broadcast*) pakietów ICMP echo (ping) z adresem IP ofiary ataku jako źródłowym. Ofiara zostanie zalana odpowiedziami ping. Atak ten jest skuteczny jedynie jeśli brzegowy dla sieci ofiary router przepuszcza ping w ukierunkowanym rozgłoszeniu, a system operacyjny stacji odpowiada na taki ping.

Fraglle attack

To atak posiadający identyczny schemat postępowania, lecz wykorzystuje usługę echo na UDP.

Land attack

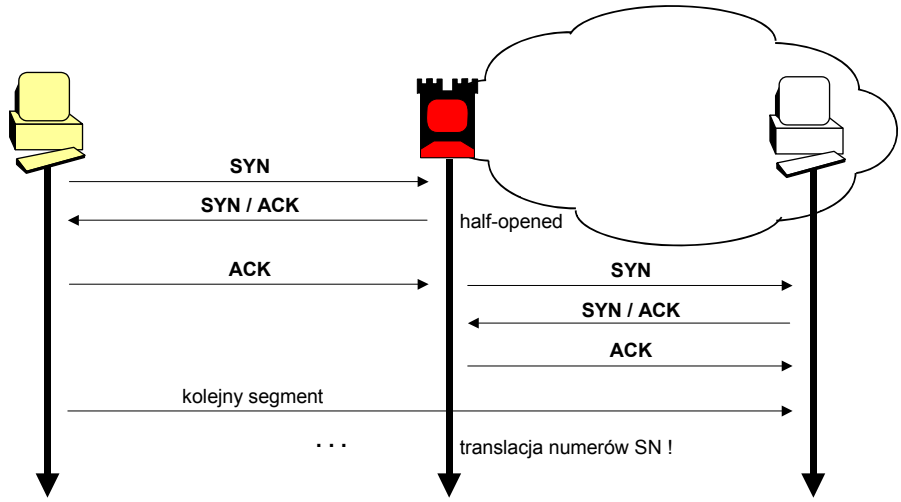
W tym przypadku, atakujący wysyła segment SYN na adres ofiary podając jej własny adres jako źródłowy i nadając ten sam numer portu źródłowego i docelowego. Stacja TCP ofiary nigdy nie zestawi połączenia zapętlając się w nieskończoność. W niektórych implementacjach może to prowadzić do jej zawieszenia.

Metody obrony przed atakami DoS/DDoS

Nie ma uniwersalnych metod obrony przez atakami odmowy dostępu. Na ogół wymagają one przygotowania przez producenta systemu operacyjnego odpowiednich poprawek (łat) i zastosowania ich w podatnym na atak systemie. Poniżej przedstawione zostaną dwa mechanizmy obrony przed atakiem SYN flood, które systematycznie zyskują coraz większą popularność.

Obrona przed atakiem SYN flood – SYN Defender

SYN Defender jest komponentem kilku kompleksowych systemów ochrony (takich jak np. CheckPoint Firewall-1). Jego ogólna koncepcja działania polega na wprowadzeniu pomiędzy atakującego i ofiarę wyspecjalizowanego obrońcę (na rysunku 14 oznaczonego na czerwono), który przejmuje wszystkie segmenty SYN skierowane do ochranianego systemu i propaguje połączenia dopiero gdy wykluczy atak (czyli, gdy dotrze do obrońcy trzeci segment nawiązania połączenia, co oznacza, że nie mamy do czynienia z atakiem SYN flood).



Rysunek 14. Schemat działania mechanizmu SYN Defender

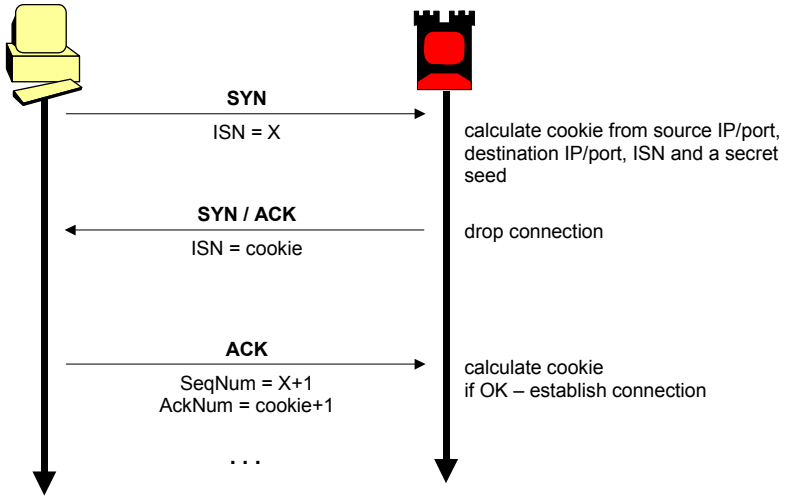
Jeśli SYN Defender rozpoznaje atak (nie doczeka się trzeciego segmentu) po prostu zapomina o parametrach połączenia, a system ochraniany nawet nie dowie się o ataku. Jeśli SYN Defender wykluczy atak, to wówczas samodzielnie zestawia nowe połączenie z systemem ochranianym, które posłuży to retransmitowania segmentów odebranych od nadawcy.

Oczywiście owo nowe połączenie nie będzie miało identycznych parametrów (np. numer ISN wybrany przez ochranianego będzie z pewnością inny, niż uprzednio zaproponowany nadawcy segmentu SYN przez obrońcę). W związku z tym, segmenty propagowane muszą być poddane konwersji parametrów (nagłówka) przy przejściu przez węzeł obrońcy).

Istotą ochrony przed atakiem jest przeniesienie punktu obrony z ofiary na zewnętrzny system, który przygotowany na okoliczność ewentualnego ataku nie pozwoli na przeciążenie siebie poprzez zużycie wszystkich zasobów. Właściwie ilość zasobów potrzebna obrońcy na obsługę połączeń jest tu minimalna, a system ofiary obsługuje wyłącznie połączenia, które nie są elementem ataku SYN flood.

Obrona przed atakiem SYN flood – SYN cookies

Inną metodą obrony przed atakiem SYN flood jest wykorzystanie dość sprytnego mechanizmu o nazwie SYN cookies. Umożliwia on realizację obrony w samym węźle potencjalnej ofiary. Przy zastosowaniu tego mechanizmu, węzeł broniący się nie musi rezerwować zasobów dla połączenia już w momencie odebrania segmentu SYN. Miast tego, węzeł ten generuje specjalną wartość, przekazywaną nadawcy segmentu SYN, i tak spreparowaną, by po ewentualnym otrzymaniu w przyszłości trzeciego segmentu (ACK), rozpoznać że jest to kontynuacja wcześniej rozpoczętego nawiązywania połączenia. Dopiero po otrzymaniu segmentu ACK rezerwowane są zasoby.



Rysunek 15. Schemat działania mechanizmu SYN cookies

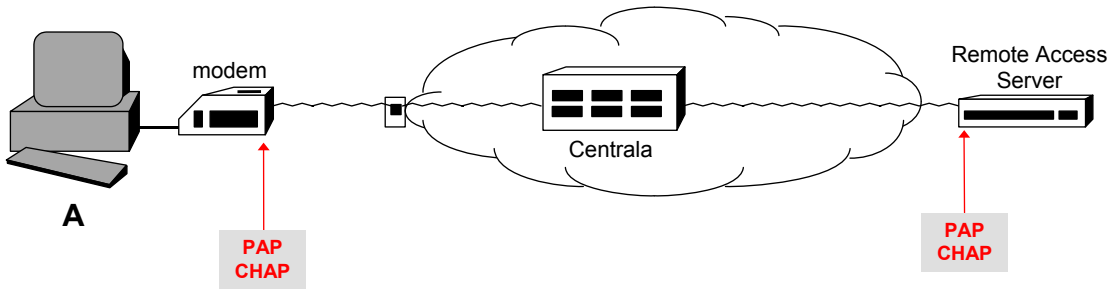
Dla rozpoznania poprawności późniejszego segmentu ACK, broniący po odebraniu segmentu SYN generuje wartość cookie uzależnioną od parametrów segmentu ACK. Wartość ta jest wpisywana następnie jako ISN do segmentu SYN/ACK, a zatem powróci (faktycznie zinkrementowana) w polu potwierdzenia w segmencie ACK, umożliwiając detekcję poprawności procedury zestawiania połączenia.

Mechanizm SYN cookies posiada niestety pewne ograniczenia, np. nie można korzystać niektórych użytecznych rozszerzeń specyfikacji protokołu TCP, np. Large Window.

Więcej informacji o SYN cookies można znaleźć pod adresem http://cr.yp.to/syncookies.html

Mechanizmy bezpieczeństwa zdalnego dostępu

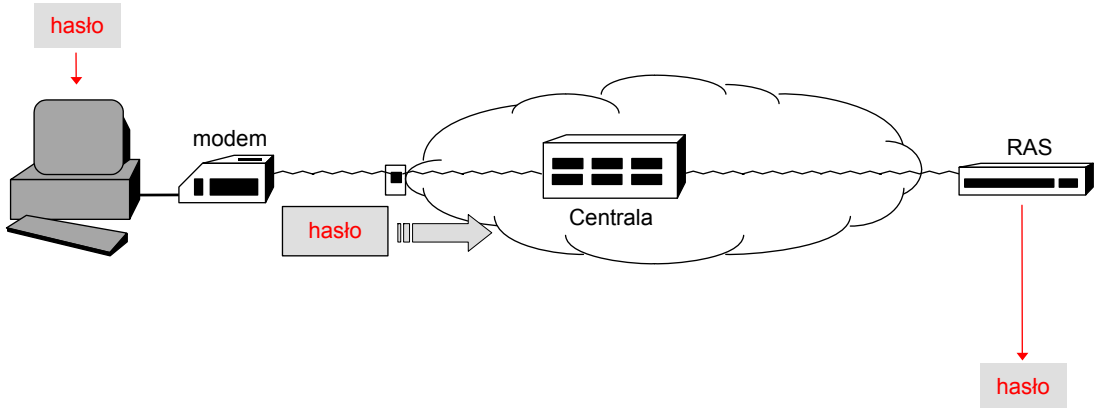
Rysunek 16 przedstawia schemat uwierzytelniania przy zdalnym dostępie do sieci komputerowej. Przyjmujemy tu scenariusz, w którym system A uzyskuje dostęp zdalny do sieci poprzez publiczną sieć operatora dostępu do Internetu. Po stronie A wykorzystywany jest adapter łącza do sieci publicznej, wbudowany w system komputerowy A lub zewnętrzny (modem telefoniczny, modem kablowy, adapter ISDN lub modem DSL). Operator posiada infrastrukturę złożoną w łączy i systemów przełączających (np. cyfrowych centralek telefonicznych). Natomiast po stronie sieci docelowej połączenie jest obsługiwane przez serwer dostępowy RAS. Serwer dostępu zezwala tylko na autoryzowane połączenia, które muszą zostać uwierzytelnione. Uwierzytelnieniu podlega na ogół adapter reprezentujący system A wobec serwera RAS. Najprostsze mechanizmy uwierzytelniania (PAP, CHAP) wykorzystują hasła.



Rysunek 16. Schemat uwierzytelniania przy zdalnym dostępie

PAP (*PPP Authentication Protocol*) RFC1334

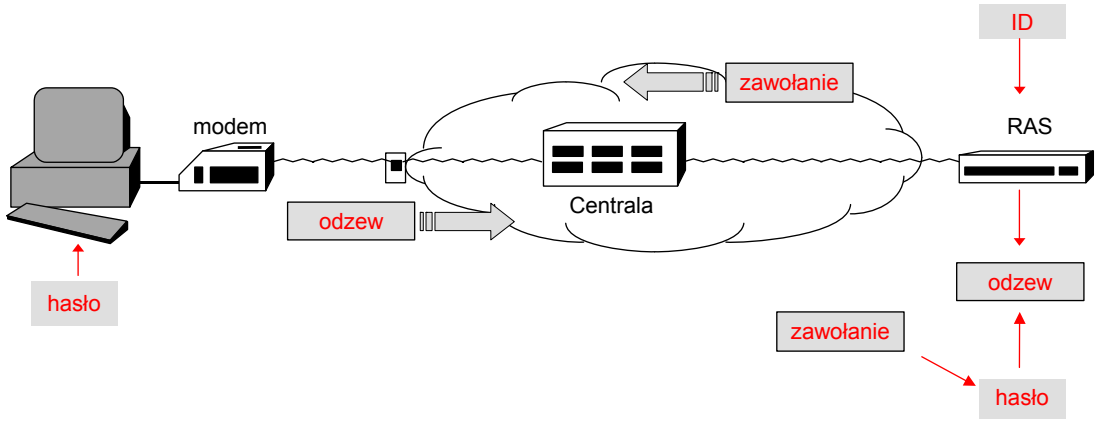
W protokole PAP, serwer RAS pyta o nazwę (ID) użytkownika, a następnie o hasło i na tej podstawie decyduje o dopuszczeniu do sieci. W tym protokole nazwa użytkownika i hasło są przesyłane tekstem jawnym! Istnieje też odmiana tego protokołu – SPAP (Shiva PAP), która stosuje proste szyfrowanie procedury uwierzytelniania.



Rysunek 17. Schemat uwierzytelniania PAP

CHAP (*Challenge Handshake Authentication Protocol*) RFC1994

W przypadku tego protokołu, RAS pyta o ID użytkownika, a następnie przesyła unikalne „zawołanie”. Klient koduje zapytanie hasłem (MD5) i odsyła jako „odzew” decydujący o dopuszczeniu do sieci.



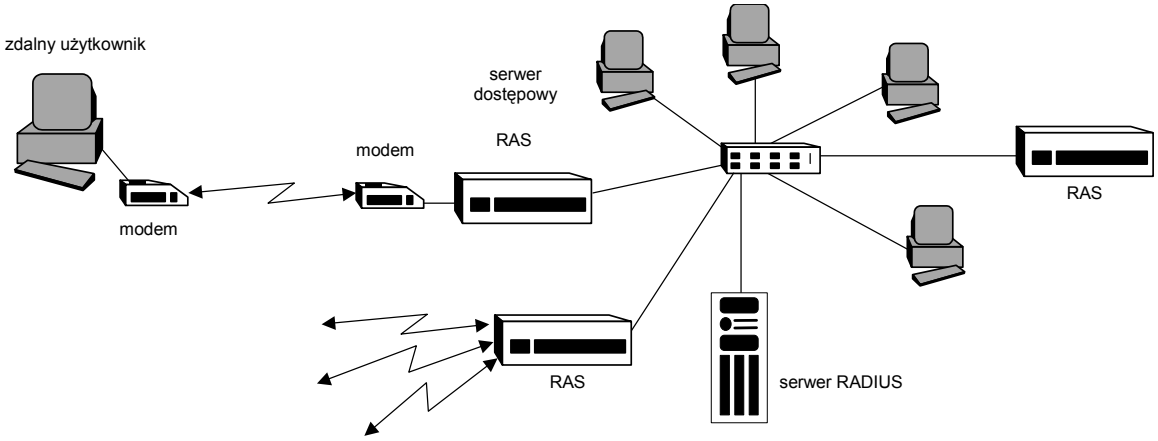
Rysunek 18. Schemat uwierzytelniania CHAP

EAP (*PPP Extensible Authentication Protocol*) RFC2284

W tym przypadku RAS wysyła kilka zapytań do uwierzytelnianego podmiotu, każdorazowo specyfikując typ żądania (np. żądanie hasła lub skrótu MD5). Oferuje tym samym możliwość korzystania z wielu protokołów uwierzytelniania bez potrzeby uprzedniego ich negocjowania. Istnieje tez możliwość dwustronnego uwierzytelniania.

Protokół RADIUS (*Remote Access Dial-In User Service*) – RFC2138

Protokół RADIUS oferuje własności AAA (Authentication + Authorization + Accounting) zdalnego dostępu i pozwala na centralizację zarządzania tymi własnościami. Pozwala na przechowywanie jednej globalnej bazy procedur i informacji uwierzytelniających, i umożliwia utrzymywanie wielu punktów dostępowych wykorzystujących tę globalną konfigurację. Ułatwia to tworzenie złożonych sieci z wieloma serwerami dostępowymi RAS. Serwer RADIUS stanowi centrum uwierzytelniania i kontroli dostępu. Natomiast punkty dostępowe RAS realizują proces uwierzytelniania na podstawie informacji pozyskanych z centralnego serwera RADIUS za pomocą protokołu RADIUS.



Rysunek 19. Konfiguracja środowiska RADIUS

Na podobnej zasadzie działają także protokoły TACACS (*Terminal Access Control – Access Control System*), XTACACS czy TACACS+.

Uwierzytelnianie stanowisk infrastruktury sieciowej

Standard IEEE 802.1x umożliwia ochronę infrastruktury sieciowej przed nieautoryzowanym dostępem poprzez centralne uwierzytelnianie poszczególnych stacji sieciowych. Przykładowo przy wykorzystaniu tego standardu przełącznik wymusza uwierzytelnienie nowo wpiętej lub właśnie uruchomionej stacji, zanim rozpocznie przełączanie pakietów przez nią wysyłanych. Do zweryfikowania danych uwierzytelniających może wykorzystać protokoły RADIUS czy TACACS+.

Bezpieczny system nazw

W 1999 zaproponowano rozszerzenie DNS o mechanizmy kryptograficznego uwierzytelniania i kontroli integralności (RFC 2535). Zaproponowano dodanie rekordów SIG zawierających podpisy cyfrowe podpisujące zbiory rekordów informacyjnych (RRset). Rolę certyfikacji pełni umieszczenie klucza publicznego w samym zbiorze. Klucz przechowują rekordy nowego typu – DNSkey. Usługa DNSsec również może służyć przechowywaniu samych kluczy publicznych dla innych celów, np. PKI. Niestety wdrożenie DNSsec wciąż napotyka trudności. Przykładowym problemem jest m.in. kwestia pełnego lub częściowego podpisywana zbiorów dla dużych domen, takich jak .com)

Narzędzia bezpieczeństwa

Niżej przedstawione zostaną dwa wybrane narzędzia ochrony środowisk sieciowych.

DoS guard

DoS guard jest nazwą osobnego narzędzia lub modułu większej aplikacji zabezpieczającej, realizującego ochronę przed atakami DoS. Funkcje DoS guard posiada większość zapór sieciowych (również osobistych), a także wiele systemów operacyjnych routerów, np. TCPintercept w CiscoIOS lub Finesse (PiX). Niektóre z dostępnych narzędzi są nawet dość zaawansowane, np. SYN defender w Checkpoint Firewall-1 lub SYN cookies w PiX-ach.

Kryptograficzne zabezpieczenie komunikacji

Jednym z najbardziej reprezentatywnych przykładów narzędzia kryptograficznej ochrony komunkacji sieciowej jest protokół SSH – Secure Shell. SSH to protokół szyfrowanej transmisji dedykowanej dla emulacji wirtualnego terminala lecz nie tylko. Protokół SSH obsługuje usługa TCP, której przydzielono port 22. W domyślnej konfiguracji zastępuje telnet, rlogin, rsh, rexec, rcp, ftp. Ponadto umożliwia tunelowanie ruchu (VPN – tryb *port forwarding*).

Protokół SSH to standard *de facto*. Istnieją jego dwie specyfikacje – SSH1 i SSH2). Dostępnych jest wiele implementacji, w tym darmowych dla większości systemów z rodziny Unix/Linux (Open SSH). Natomiast dla systemów MS Windows, MacOS dostępnych jest wielu klientów protokołu SSH.

SSH1	SSH2
DES	Arcfour
3DES	3DES (domyślny)
IDEA (domyślny)	Twofish
Blowfish	Blowfish
RC4 128b	RC4 128b
RSA	DSA
	CAST 128
	D-H key exchange

Rysunek 20. SSH – wykorzystywane algorytmy krytpograficzne.

Implementacje mogą domyślnie używać inne algorytmy niż wskazane w specyfikacji protokołu (np. OpenSSH)

SSH oferuje różnorodne metody uwierzytelniania, m.in. tradycyjne – hasłem konta systemu zdalnego, kryptograficzne – zapytanie odzew z kluczem publicznym i prywatnym RSA, czy wykorzystanie zewnętrznych systemów uwierzytelniania, jak Kerberos lub S/Key. Istnieją implementacje wykorzystujące tokeny elektroniczne.

Pytania problemowe

1. W bieżącym module stwierdzono, iż często filtry przetwarzają właściwie tylko pierwszy fragment pakietu ponieważ informacje niezbędne do przeprowadzenia filtracji znajdują właśnie w pierwszym pakiecie. Spróbuj wyjaśnić dlaczego.

2. Dlaczego akurat usługi r* są szczególnie upodobanym celem ataków TCP spoofing?