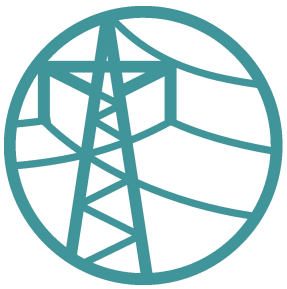


WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ

Motyka Beniamin

Bezpieczeństwo IT dla firm - opis i implementacja

Praca dyplomowa inżynierska

Opiekun pracy:
dr Michał Piętal

Rzeszów, 2021

Spis treści

1. Wprowadzenie

W dzisiejszych czasach śmiało można stwierdzić, iż Internet stał się ważną częścią ludzkiego istnienia. Niewątpliwy wpływ na ten stan rzeczy miała pandemia COVID-19 - sprawiła ona bowiem, że pewne dziedziny życia, takie jak dydaktyka czy praca wykonywana umysłowo, przeszły swoistą transformację. Miejsca, w których spotykali się studenci wraz z wykładowcami, czy pracownicy w biurze, stały się puste. Zastąpiła je komunikacja zdalna – przez Internet.

Fakt, iż ludzkość została zmuszona, by przenieść znaczną część swojego funkcjonowania w sieć, niesie ze sobą poważne konsekwencje. Szybkie – jak do tej pory – tempo rozwijania się technologii informatycznych stało się nieporównywalnie bardziej dynamiczne, a co się z tym wiąże, obecne w sieci liczne zagrożenia stały się coraz powszechniejsze i trudniejsze w identyfikacji.

[coś tu jeszcze będzie]

1.1. Cel i zakres pracy

Celem niniejszej pracy inżynierskiej jest wyeksponowanie, opis oraz implementacja najbardziej pospolitych zagrożeń i luk bezpieczeństwa w Internecie nie tylko dla zwykłych użytkowników, ale również dla małych i średnich przedsiębiorstw. Dzięki temu, że powyższa idea zostanie zrealizowana w formie aplikacji e-learningowej, istnieje realna szansa na zwiększenie świadomości społecznej, edukację oraz poprawę zabezpieczeń systemów teleinformatycznych i infrastruktury sieciowej. Zakresem pracy są takie zagadnienia jak:

- Przegląd, dokumentacja i implementacja zagrożeń i luk bezpieczeństwa w sieci.
- Stworzenie aplikacji e-learningowej przy użyciu technologii opisanych w kolejnym rozdziale. Użytkownicy będą mogli wziąć udział w interaktywnej prezentacji każdego z zagrożeń.
- Zasugerowanie potencjalnych rozwiązań na opisane cyberzagrożenia.
- Przedstawienie wniosków i implikacji płynących z powyższych.

[coś tu jeszcze będzie]

2. Technologie wykorzystane w aplikacji

tekst

2.1. Javascript

tekst

2.1.1. React.js

tekst

2.1.2. Node.js

tekst

2.2. GraphQL

tekst

2.3. MongoDB

tekst

3. Zagrożenia bezpieczeństwa

tekst

3.1. Phishing - opis

tekst

3.2. Phishing - przykład implementacji

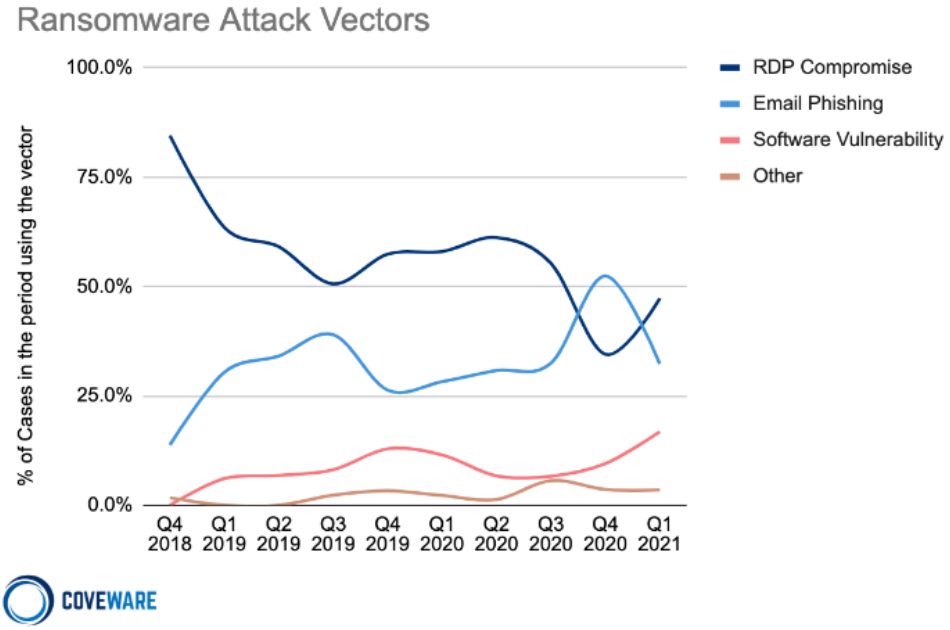
tekst

3.3. Ransomware - opis

Ransomware to typ złośliwego oprogramowania, którego celem jest zablokowanie dostępu do komputera osobistego. Kiedy to złośliwe oprogramowanie dostanie się na urządzenie jako załącznik w wiadomości e-mail lub poprzez sieć, rozpoczyna proces szyfrowania asymetrycznego wszystkich plików. Czas trwania tej procedury zależy między innymi od wybranego algorytmu szyfrowania, jednak może wykonać się w czasie zaledwie pięciu godzin [7]. Zazwyczaj ten czas jest znacznie dłuższy, gdyż przed rozpoczęciem procesu wymagany jest dokładny rekonesans systemu, w którym atakujący się znajduje. Tym sposobem użytkownik końcowy traci możliwość odczytu danych na swoim urządzeniu, a do odszyfrowania plików, wymagany jest klucz posiadany przez atakującego.

Ten rodzaj szkodliwego oprogramowania jest szczególnie niebezpieczny dla przedsiębiorstw, gdyż utrata ważnych dokumentów czy danych finansowych, może się wiązać z poważnymi konsekwencjami. Celem ransomware nie jest usunięcie lub kradzież plików, ale zablokowane ich, i oczekiwanie na ewentualną spłatę okupu ze strony ofiary.

Sposoby, poprzez które ransomware dostaje się do przedsiębiorstw, przedstawione są na poniższym rysunku, przygotowanym przez organizację CoveWare [6], która specjalizuje się w przeciwdziałaniu tego typu oprogramowaniu:



Rysunek 3.1: Wektory ataku ransomware

Jak można zauważyć, wiodącym źródłem pozyskania tego złośliwego oprogramowania jest Remote Desktop Protocol (RDP) [5] - protokół umożliwiający połączenie się z urządzeniem, oraz przejęcie nad nim pełnej kontroli, używany w firmach jako narzędzie do konfiguracji systemu dla pracowników. Zaraz za nim znajdują się maile phishingowe, opisane w poprzednim rozdziale, a następnie różnego rodzaju luki w oprogramowaniu.

Najlepszą metodą przeciwko tego typu złośliwemu oprogramowaniu jest systematyczne wykonywanie archiwizacji danych. W tym wypadku, jeśli ransomware trafi do komputera i zaszyfruje wszystkie pliki, pozostaje jedynie odtworzyć kopię zapasową.

[coś tu jeszcze będzie]

3.4. Ransomware - przykład implementacji

tekst

3.5. Keylogger - opis

Keyloggery występują zazwyczaj w formie złośliwego, ukrytego oprogramowania. Nie są widoczne na pierwszy rzut oka dla użytkownika i w zdecydowanej większości przypadków działają w tle, często podszywając się pod inną aplikację, tym samym maskując swoją obecność.

Podstawowe właściwości tego typu złośliwego programu można opisać jako przejmowanie kontroli nad procedurami związanymi z obsługą klawiatury systemu operacyjnego, na którym się znajduje.

Głównym celem tego oprogramowania jest zbieranie danych o tym, jakie klawisze na klawiaturze zostały naciśnięte przez użytkownika, a następnie okresowe wysyłanie zebranych informacji do atakującego. Posiadając wiedzę na temat tego, co zostało wpisane na urządzeniu, można bez problemu uzyskać dostęp do wrażliwych informacji takich jak prywatna korespondencja czy poufne dane. Do bardziej zaawansowanych funkcji należy między innymi przesyłanie zrzutów ekranu, rejestrowanie historii otwieranych programów i przekazywanie tych informacji dalej.

Keyloggery oprócz formy programowej istnieją również jako osobne urządzenia, które podłączane są do jednostki zazwyczaj poprzez interfejs USB. Mogą także występować jako przejściówka pośrednicząca pomiędzy klawiaturą a złączem USB komputera.

Sposobem na unikanie tego typu oprogramowania jest przede wszystkim systematyczne sprawdzanie uruchomionych procesów, ale także używanie odpowiedniego antywirusa.

3.6. Keylogger - przykład implementacji

tekst

3.7. Wstrzyknięcie SQL - opis

tekst

3.8. Wstrzyknięcie SQL - przykład implementacji

tekst

3.9. DDOS - opis

tekst

3.10. DDOS - przykład implementacji

tekst

3.11. Ataki XSS - opis

Cross-site scripting (XSS) jest jedną z najbardziej powszechnych podatności współczesnych aplikacji webowych.

Opisując XSS nie sposób nie wspomnieć o Regule tego samego pochodzenia (Same-origin policy) [2]. Jest to jeden z wielu fundamentalnych mechanizmów bezpieczeństwa, zaimplementowany w każdej przeglądarce internetowej. Nie pozwala on żadnej stronie na podjęcie akcji lub odczytania zawartości innej strony, na przykład w dwóch kartach przeglądarki. W związku z tym, wszystko, co się dzieje na stronie internetowej otwartej przez użytkownika, jest izolowane, i nie będzie miało wpływu na pozostałe otwarte strony.

Cała struktura strony internetowej zakodowana językiem HTML może być zmieniana poprzez JavaScript, używając DOM API [3]. Jako rezultat, prosty skrypt może całkowicie zmienić zawartość, wygląd, a przede wszystkim funkcjonalność strony inter-

netowej.

Ciasteczkami [4] nazywa się niewielkie informacje wysyłane przez serwer do przeglądarki internetowej urządzenia końcowego. Służą temu, by zapisać obiekty danych w przeglądarce, które przy ponownym odwiedzeniu strony, mogą być przesłane do tego samego serwera, z którego przyszły. W związku z tym, przy odwiedzaniu różnorakich stron wymagających autoryzacji, użytkownik nie musi się za każdym razem logować, gdyż wszystkie potrzebne dane są zawarte w ciasteczkach, które przesyłane są razem z żądaniem.

XSS opiera się głównie na wstrzyknięciu do strony internetowej złośliwego skryptu, który, dla przykładu, może odczytać ciasteczka użytkownika lub inne poufne informacje, które przechowuje przeglądarka, wysłać je do atakującego, aby ten – używając zapisanych w ciasteczkach danych – mógł zalogować się na konto użytkownika, który nieświadomie uruchomił dany skrypt. Pomimo wielu zabezpieczeń, które są wbudowane w przeglądarki, te nie są w stanie odróżnić czy dany skrypt jest złośliwy, czy nie - dlatego też odporność aplikacji internetowej na tego typu atak stoi przede wszystkim po stronie programistów.

Najczęstszym miejscem, w którym można spotkać tę podatność są formularze, do których użytkownik wpisuje jakąś treść, która następnie jest wyświetlana. Jeśli treść nie zostanie odpowiednio zwalidowana, może dojść do sytuacji, w której to użytkownik wstrzyknie złośliwy skrypt.

[coś tu jeszcze będzie]

3.12. Ataki XSS - przykład implementacji

tekst

3.13. Path Traversal - opis

"Path Traversal"jest to nazwa dla powszechnej luki bezpieczeństwa aplikacji Internetowych, której niedopatrzenie, w procesie tworzenia oprogramowania, może skutkować w wycieku wrażliwych danych z serwera, na którym umieszczona jest aplikacja.

Dzięki językom programowania działających po stronie serwera, takim jak PHP, zewnętrzne skrypty i pliki mogą być dołączane do aplikacji w sposób dynamiczny. Krytycznym elementem w tego typu funkcjonalnościach jest odpowiednio zaimplementowana logika dołączania plików oraz walidacja danych wejściowych, gdyż w przeciwnym

wypadku, atakujący może odczytywać zawartość plików lokalnych, jak i zdalnych. Celem tej podatności jest zlokalizowanie i odpowiednie wykorzystanie parametrów przekazywanych do aplikacji, poprzez które dynamicznie dołączane są różne skrypty.

Poniższy przykład prezentuje dołączenie pliku 'pl.php' jako parametr, celem ustawienia tekstu na stronie w języku polskim:

`http://samplepage.com/index.php?lang=pl.php`

Może to świadczyć o tym, że przyłączenie pliku do w kodzie źródłowym aplikacji wykonywane jest w następujący, przykładowy sposób:

```
1  include($_GET['lang']);
```

Listing 1: Listing programu PHP

Mając na uwadze fakt, że do aplikacji dynamicznie dołączany jest skrypt przekazywany w parametrze 'lang', atakujący może dokonać próby otworzenia zupełnie innego pliku, niż było to zakładane:

`http://samplepage.com/index.php?lang=../../../etc/passwd`

W rezultacie, zamiast wypisania tekstu w języku polskim, przekazana zostanie zawartość pliku '/etc/passwd', w którym to znajdują się informacje o wszystkich użytkownikach w systemie.

Aby zapobiec tej luce, zamiast wykluczać lub usuwać dane ciągi znaków należy odpowiednio sprawdzać dane wejściowe i zezwalać jedynie na wybrane znaki lub ciągi znaków, na przykład z wykorzystaniem wyrażeń regularnych.

3.14. Path Traversal - przykład implementacji

tekst

3.15. Spoofing - opis

tekst

3.16. Spoofing - przykład implementacji

tekst

4. Podsumowanie i wnioski końcowe

Literatura

[1] Securitum. Bezpieczeństwo aplikacji webowych

[2] https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy

[3] https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model.

[4] https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies.

[5] https://docs.microsoft.com/en-us/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol

[6] https://www.coveware.com/blog/ransomware-attack-vectors-shift-as-new-software-vulnerability-exploits-abound

[7] https://thedfirreport.com/2020/10/18/ryuk-in-5-hours/

POLITECHNIKA RZESZOWSKA im. I. Łukasiewicza Rzeszów, 2021
Wydział Elektrotechniki i Informatyki

STRESZCZENIE PRACY DYPLOMOWEJ INŻYNIERSKIEJ

BEZPIECZEŃSTWO IT DLA FIRM - OPIS I IMPLEMENTACJA

Autor: Motyka Beniamin, nr albumu: EF-160780
Opiekun: dr Michał Piętal
Słowa kluczowe: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po polsku

RZESZOW UNIVERSITY OF TECHNOLOGY Rzeszow, 2021
Faculty of Electrical and Computer Engineering

BSC THESIS ABSTRACT

TEMAT PRACY PO ANGIELSKU

Author: Motyka Beniamin, nr albumu: EF-160780
Supervisor: (academic degree) Imię i nazwisko opiekuna
Key words: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po angielsku