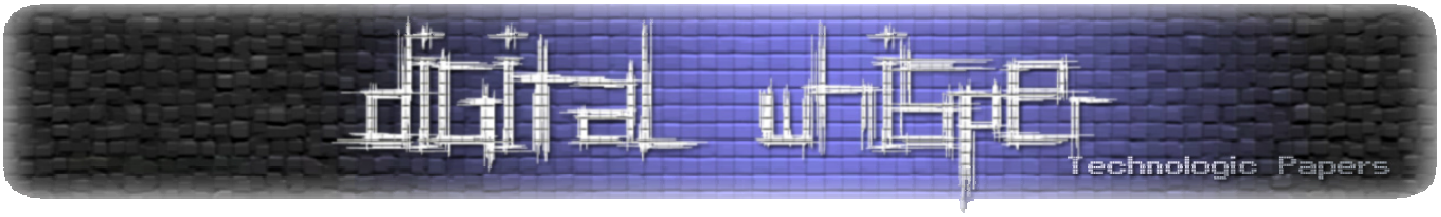


HTTP Fingerprints

מאת אפיק קסטיאל (cp77fk4r)

ישנן דרכים רבות לאסוף מידע על שרתי HTTP. אחת הדרכים המוכרות ביותר היא בעזרת איסוף ה- HTTP Fingerprints שלהם. Fingerprint הוא שם כולל לערך, תגובה או פעולה מסויימת הייחודית עבור שירות מסויים אשר בעזרתו נוכל לזהות את המוצר או את גירסתו.

HTTP Fingerprints הוא שם כולל לכלל ה-Banners הקיימים בפרוטוקול ה-HTTP וניתן לאסוף אותן בקלות על ידי מעקב אחרי ה-Request וה-Response בין תוכנת הלקוח לבין השרת. רוב שירותי ה-HTTP אומנם מיישמים באופן דומה את פרוטוקול ה-HTTP, אך לא באופן זהה לחלוטין ובכולם אפשר למצוא "חתימה" או "טביעת אצבע" המאפשרת לתוקפים לזהות אותם בעזרתה.

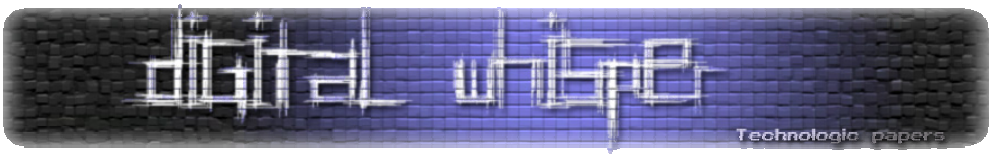
למה, בתור בעלי השרת, חשוב לנו אם תוקפים יוכלו לזהות את גרסאות ומאפייני השירותים שלנו? ובכן, במידה וקיימות פרצות מוכרות לשירותים שלנו (וכל יום מתגלות פרצות חדשות), אותם תוקפים יוכלו לאתר אותן ביתר קלות ולא יזדקקו למחקר מפרך. כיום קיימים כל כך הרבה אתרים המציעים מנגנוני חיפוש ורשימות ארוכות של חולשות ופרצות שנמצאו במגוון רחב של שירותים. קיימים אף מספיק כלים המבצעים את הסריקות הנ"ל באופן אוטומטי.

במאמר זה נתמקד בשני שרתים מרכזיים, הראשון הוא-IIS והשני הוא-Apache, שניהם שרתי ה-HTTP הנפוצים ביותר כיום. בנוסף, נסקור דרכים שונות למנוע מאותם שרתים לפלוט HTTP Fingerprints בכדי להקשות על אופן הזיהוי של אותם שרתים.

אילו נתונים אפשר לזהות ע"י איסוף ה-Fingerprints?

- סוג השרת
- גירסת השרת
- טכנולוגיות/Frameworks המותקנות על השרת וגירסאותיהן
- מודולים בהם רץ השרת

כאשר שירותים חושפים מידע המאפשר לתוקף לגלות פרטים הממקדים אותו, קוראים לחשיפה "Information Leakage" או "Information Disclosure".



מספר דוגמאות

בבית יש לי XAMPP 2.5 (עליו שרת Apache/2.2.9) מותקן על המחשב ו-Windows Server 2003 עם IIS 6.0 המריץ WebDav על מכונה וירטואלית. נבחן את ה-Apache לאחר התקנת ברירת מחדל מבלי לשנות שום קונפיגורציה. אם נשלח אליו HTTP HEAD REQUEST (בקשה המורה לשרת לפלוט את תכני ה-HTTP HEADERS שלו) באופן הבא:

```
HEAD /index.php HTTP/1.1
Host: localhost
```

השרת יפלוט לנו:

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 15:15:48 GMT
Server: Apache/2.2.9 (Win32) DAV/2 mod ssl/2.2.9 OpenSSL/0.9.8h
mod_autoindex_color PHP/5.2.6
X-Powered-By: PHP/5.2.6
Content-Type: text/html
```

שימו לב לקטע המודגש באדום. לפי שדה ה-SERVER ניתן להבין כי אנו עומדים מול שרת Apache, שגירסתו היא 2.2.9 ומערכת ההפעלה עליו הוא רץ היא מסוג Windows 32 bit, לא מצויינת הגרסא. בנוסף השרת מחזיר לנו את המודולים שבהם הוא תומך, את גרסאותיהם ואת גירסאת ה-PHP שבה הוא תומך: 5.2.6.

דוגמא נוספת: ה-RESPONSE הבא נאסף מאתר של מגזין מאוד מעניין בנושא האקינג וטכנולוגיה (לא, לא Digital Whisper):

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 16:03:52 GMT
Server: Apache/2.2.3 (Debian) DAV/2 PHP/5.2.0-8+etch13
X-Powered-By: PHP/5.2.0-8+etch13
Connection: close
Content-Type: text/html; charset=UTF-8
```

ניתן לראות שגם הפעם מדובר בשרת Apache, גירסא 2.2.3. כמו כן, הפצת הלינוקס שעליה הוא רץ היא Debian. לפי שדה ה-X-Powered-By נבחין כי גירסאת ה-PHP שבה הוא תומך היא 5.2.0-8+etch13 (גירסאת 5.2.0 שעברה כמות נכבדת של הטלאות).

דוגמא אחרונה שנציג היא RESPONSE של שרת IIS שמריץ אתר תוכן ישראלי מאוד מוכר:

```
HTTP/1.1 200 OK
Connection: keep-alive
Date: Wed, 25 Nov 2009 16:26:10 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Content-Type: text/html
```

כפי שניתן לראות, אכן מדובר בשרת IIS, גירסא 6.0 שהפעם תומך בטכנולוגית .NET., לפי שדה ה--X AspNet-Version אנחנו יכולים להסיק כי גרסאת ה-.NET Framework שמותקנת עליו היא 2.0.50727.

לאחר הצגת כל הנתונים הללו נשאלת השאלה כמה המידע הזה באמת רלוונטי ועד כמה אנחנו אמורים לחשוש מפניו בתור מנהלי השרת. אינפורמציה זו יכולה להיות לא רלוונטית בכלל ויכולה להיות אחת הנקודות החלשות במערכת שלנו. יש לזכור כי הפירצה עצמה היא אינה ה-Fingerprint, איסוף ה-Fingerprint היא רק דרך שבאמצעותה ניתן לאתר מערכת רגישה. לדוגמא, ב-RESPONSE האחרון שהצגנו- כמעט ולא משנה באיזה אופן יכתבו את מערכת האתר שתרוץ על השרת, כמעט בכל המקרים היא תהיה פגיעה למתקפת XSS-UTF7. בשל שני גורמים:

- זאת חשיפה שקיימת בכל .NET Framework. בגירסאות 2.x.x (ברגע שמנסים לגשת לעמוד .NET שלא קיים)
- אחד הפתרונות ל-XSS-UTF7 הוא להגדיר את ה-Content-Type, באופן הבא:

```
Content-Type: text/html; charset=UTF-8
```

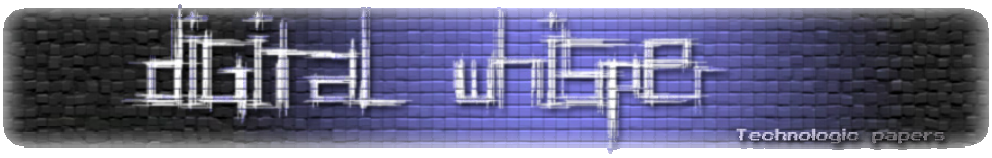
וכמו שאנחנו רואים, כאן הוא מוגדר לנו רק כ:

```
Content-Type: text/html
```

אין משמעות הדבר כי בוודאות קיימת פירצה שאפשר לנצל אותה על השרת, אך במקרה כזה כדאי מאוד לבדוק.

חוץ ממידע על נתוני השרת ומאפייניו, ניתן גם לשלוף מידע על תצורתו, על המתודות המורשות עליו וכו': שליחת HTTP OPTIONS REQUEST (שאילתה המבקשת מהשרת לפלוט את המתודות שבהן הוא תומך), לשרת ה-Apache 2.2.9:

```
OPTIONS / HTTP/1.1
Host: localhost
```



תניב את התגובה הבאה:

```
HTTP/1.1 200 OK
Date: Thu, 26 Nov 2009 08:44:31 GMT
Server: Apache/2.2.9 (Win32) DAV/2 mod_ssl/2.2.9
OpenSSL/0.9.8hmod_autoindex_color PHP/5.2.6
Allow: GET,HEAD,POST,OPTIONS,TRACE
Content-Length: 0
Content-Type: httpd/unix-directory
```

בשדה ה-Allow ניתן לראות כי חוץ מהמתודות HEAD ו-OPTIONS שהשתמשנו בהן, השרת תומך גם במתודות GET ו-POST בכדי לשלוח ולקבל מידע מהלקוח. בנוסף, השרת תומך במתודת TRACE (מתודה המשמשת כ-LOOP-BACK לצרכי Debugging בעיקר). בעבר נעשה שימוש במתודה הזאת במתקפה בשם XST (Cross Site Tracing) בכדי לעקוף את מנגנון ה-HTTPOnly אשר בא למנוע מתקפות לגניבת ה-Cookies של המשתמשים על ידי חשיפות כגון XSS (Cross Site Scripting).

שליחת HTTP OPTIONS REQUEST לשרת ה-Windows Server 2003 שלנו תוביל לתגובה הבאה:

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 26 Nov 2009 08:36:17 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
MS-Author-Via: DAV
Content-Length: 0
Accept-Ranges: none
DASL: <DAV:sql>
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE,
MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK, SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK
Cache-Control: private
```

מהתבוננות במתודות הנתמכות על ידי השרת נסיק כי מותקן עליו שירות WebDav (שירות הרץ על גבי פרוטוקול ה-HTTP המאפשר עבודה משותפת על משאבים הנמצאים על שרת מרוחק). בכדי לאמת את ההנחה שלנו, אפשר לנסות להשתמש במתודה PROPFIND:

```
PROPFIND / HTTP/1.1
Host: localhost
Content-Length: 0
```

במידה והשירות מופעל, נקבל תגובת 207 בסיגנון הבא:

```
HTTP/1.1 207 Multi-Status
Date: Thu, 26 Nov 2009 10:08:57 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Content-Type: text/xml
Content-Length: 747

<?xml version="1.0"?><a:multistatus xmlns:b="urn:uuid:c2f41010-65b3-
11d1-a29f-00aa00c14882/" xmlns:c="xml:"
xmlns:a="DAV:"><a:response><a:href>http://localhost/</a:href><a:propsta
t><a:status>HTTP/1.1 200 OK</a:status><a:prop><a:getcontentlength
b:dt="int">0</a:getcontentlength><a:creationdate
b:dt="dateTime.tz">2009-11-
25T19:44:50.446Z</a:creationdate><a:displayname>/</a:displayname><a:get
etag>"e0e35cc076eca1:23a"</a:getetag><a:getlastmodified
b:dt="dateTime.rfc1123">Wed, 25 Nov 2009 19:44:50
GMT</a:getlastmodified><a:resourcetype><a:collection/></a:resourcetype>
<a:supportedlock/><a:ishidden
b:dt="boolean">0</a:ishidden><a:iscollection
b:dt="boolean">1</a:iscollection><a:getcontenttype/></a:prop></a:propst
at></a:response></a:multistatus>
```

במידה ולא, נקבל שגיאה 501:

```
HTTP/1.1 501 Not Implemented
Content-Length: 0
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 26 Nov 2009 10:11:27 GMT
```

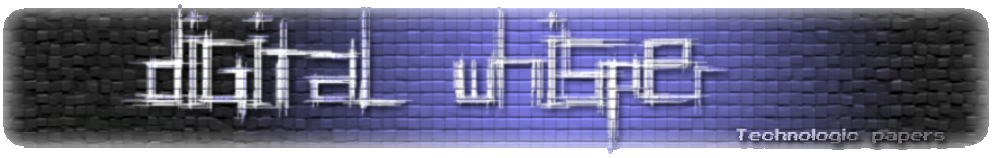
בעזרת המתודה PROPFIND ניתן לשלוף מידע על שאר המשאבים המוגדרים תחת אותו השירות.

באמצע מאי השנה (2009), בחור בשם Kingcope (Nicolaos Rangos) פרסם מאמר תחת הכותרת:

"Microsoft IIS 6.0 WebDAV Remote Authentication Bypass"

במאמר זה הוא מציג חשיפה בשירות ה-WebDav הרץ על שרתי IIS 6.0 המאפשרת לעקוף את מנגנון האותנטיקציה הקיים בשירות, בעזרתה אפשר לגשת לקבצים מוגבלים, להציג ולהעלות קבצים לשרת ובעקבותיה הומלץ פשוט לא להשתמש בשירות "עד להודעה חדשה". ליותר מידע:

http://seclists.org/fulldisclosure/2009/May/att-134/IIS_Advisory_pdf.bin



התחלת עבודה

כל השרתים מגיעים עם ממשקי ניהול/קבצי קונפיגורציה שנועדו בין היתר גם לקבוע אילו באנרים יחשפו בתגובות ה-HTTP, בחלק הזה של המאמר נסביר צעד צעד איך אפשר לבטל את רוב הבאנרים. חשוב לזכור שביטול הבאנרים והסתרת "טביעות האצבע" לא יסגרו את הפרצות בשרת, אך הדבר יקשה על תוקפים לזהות את מאפייני השרת והשירותים הרצים בו.

Obfuscation לשרת ה-IIS 6.0

הורדת ה-Server Banner

בכדי לבטל את שליחת ה-"Server Banner" בכל HTTP RESPONSE כמו זה:

```
HTTP/1.1 200 OK
Content-Length: 1433
Content-Type: text/html
Content-Location: http://localhost/iisstart.htm
Last-Modified: Fri, 21 Feb 2003 16:48:30 GMT
Accept-Ranges: bytes
ETag: "0c3110c9d9c21:23b"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 26 Nov 2009 10:48:06 GMT
Connection: close
```

עד ה-IIS 5.0 היה אפשר לגשת למפתח הבא ולהציב 1 בערך DisableServerHeader:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters
```

ב-IIS 6.0 אפשרות זו ירדה. אך יש פתרונות אחרים לצורך העניין, הורידו את הקובץ הבא:

http://www.asp101.com/articles/wayne/pryingeyes/download/XMask.zip

(זה קובץ DLL ל-ISAPI, שמרו אותו במיקום: %windir%\system32\inetsrv).

את הקובץ הנ"ל יש לטעון ל-ISAPI Filters, בצורה הבאה:

- כנסו לממשק הניהול של השרת, ושם בחרו ב-Administrative Tools.
- כנסו ל-Internet Information Services (IIS) Manager.
- ב-Web Sites כפתור שמאלי על תיקית האתר שלכם ובחירה ב-Properties.
- כנסו ל-ISAPI Filters וביחרו ב-Add.
- ב-Filter name כיתבו משהו כמו "Server Header Remover". וב-Executable הכניסו את הקובץ שהורדתם:

%windir%\system32\inetsrv\XMask.dll

- לחצו Apply וסגרו את התפריט.

בדיקה:

```
HEAD / HTTP/1.0
Host: localhost
```

התגובה שנקבל תהיה:

```
HTTP/1.1 200 OK
Content-Length: 1433
Content-Type: text/html
Content-Location: http://localhost/iisstart.htm
Last-Modified: Fri, 21 Feb 2003 16:48:30 GMT
Accept-Ranges: bytes
ETag: "0c3110c9d9c21:274"
X-Powered-By: ASP.NET
Date: Thu, 26 Nov 2009 11:39:08 GMT
Connection: close
```

שימו לב שה-RESPONSE לא כלל את ה-Server Banner.

למתעניינים בקוד הפילטר , אפשר להורידו מכאן:

http://www.asp101.com/articles/wayne/pryingeyes/download/XMaskSrc.zip

הורדת ה-X-Powered-By Banner

כדי להוריד את ה-X-Powered-By Banner יש לפעול כך:

- כנסו לממשק הניהול של השרת, ושם ביחרו ב-Administrative Tools.
- כנסו ל-Internet Information Services (IIS) Manager.
- ב-Web Sites כפתור שמאלי על תיקית האתר שלכם ובחירה ב-Properties.
- שם כנסו ל-HTTP Headers ותורידו את X-Powered-By: ASP.NET.
- לחצו Apply וסגרו את התפריט.

בדיקה:

```
HEAD / HTTP/1.0
Host: localhost
```

התגובה שנקבל תהיה:

```
HTTP/1.1 200 OK
Content-Length: 1433
Content-Type: text/html
Content-Location: http://localhost/iisstart.htm
Last-Modified: Fri, 21 Feb 2003 16:48:30 GMT
Accept-Ranges: bytes
ETag: "0c3110c9d9c21:287"
Date: Thu, 26 Nov 2009 11:51:17 GMT
Connection: close
```

שימו לב שעכשיו ה-RESPONSE גם לא כלל את ה-X-Powered-By Banner.

הורדת ה-X-AspNet-Version

כמו שראינו בדוגמאות בתחילת המאמר, במספר מקרים השרת גם יכלול את גירסאת ה-ASP שהוא מריץ ע"י השדה X-AspNet-Version.

בכדי להפטר ממנו, פשוט יש להוסיף את השורה:

```
<httpRuntime enableVersionHeader="false" />
```

תחת התגית <system.web> לקובץ ה-Web.config בתיקיה הראשית, או לגשת לקובץ-

```
%windir%\Microsoft.NET\Framework\[FWversion]\CONFIG\machine.config
```

ותחת התגית <httpRuntime> להגדיר כך:

```
enableVersionHeader="false"
```

ביטול התמיכה ב-HTTP OPTIONS/TRACE Methods

בכדי לקבוע אילו מתודות אנו רוצים לאפשר בשרת ואילו לא, נוכל לעשות שימוש בעוד ISAPI Filter מפורסם, בשם URLScan, אותו ניתן להוריד מכאן:

http://download.microsoft.com/download/c/7/a/c7a411ed-1c0f-48c1-90e5-6d3a1ca054c1/urlscan_v31_x86.msi

לאחר ההתקנה הוא טוען את עצמו לשרת. ה-URLScan מגיע מקונפג לעבודה סבירה מול שרת, אך מי שרוצה לקבוע בעצמו את ההגדרות או את הבאנרים שיכנס לקובץ URLscan.ini, שמיקום ברירת המחדל שלו הוא:

```
%windir%\system32\inetsrv\urslscan\
```

ושם יוכל לקבוע אילו מתודות הוא מעוניין לאפשר על השרת, באופן הבא, יש לוודא שמוגדר UseAllowVerbs=1 ואז להכניס רק את המתודות שהוא מעוניין לאפשר מתחת לתגית: [AllowVerbs], לדוגמא:

```
[AllowVerbs]
GET
POST
HEAD
```

בנוסף למאפיינים שהצגנו, משתמשים ב-URLScan גם לצרכים אחרים, כגון סינון שאילתות המכילות תווים נפוצים בתקפות כגון SQL Injection ו-Cross Site Scripting. הרעיון ב-ISAPI Filter הוא שהוא יושב לפני שרת ה-IIS ומתפקד כמעין IDS, כך שגם אם השרת אכן תומך במתודות מסויימות וה-ISAPI Filter חוסם אותן- בקשות HTTP העושות שימוש במתודות אלה לא יגיעו אליו מפני שהם לא יעברו את ה- ISAPI Filter.

Obfuscation לשרת ה-Apache2

הורדת ה-Server Banner

בכדי לבטל את שליחת ה-"Server Banner" בכל HTTP RESPONSE כזה:

```
HTTP/1.1 200 OK
Date: Thu, 26 Nov 2009 22:32:44 GMT
Server: Apache/2.2.9 (Win32) DAV/2 mod_ssl/2.2.9
OpenSSL/0.9.8h mod autoindex color PHP/5.2.6
X-Powered-By: PHP/5.2.6
Content-Type: text/html
```

בגרסאות Apache 2.x, ראשית יש לאפשר את המוד של mod_headers ב-httpd.conf על ידי הורדת הסולמית לפני השורה:

```
LoadModule headers_module modules/mod_headers.so
```

לאחר מכן, יש להוסיף בסוף הקובץ את השורה:

```
ServerTokens Prod
```

נבצע בדיקה:

```
HTTP/1.1 200 OK
Date: Thu, 26 Nov 2009 23:23:41 GMT
Server: Apache
X-Powered-By: PHP/5.2.6
Content-Length: 0
Content-Type: text/html
```

כמו שתוכלו להבחין הורדנו את כלל הפירוט של השרת אך עדיין מצויין כי השרת הוא שרת Apache. השרת אינו תומך בהורדת כלל הבאנר, ולכן הפתרון הוא להוריד את הקוד-מקור של ה-httpd.h לערוך אותו ולקמפל מחדש. קוד המקור ניתן להורדה כאן:

http://www.temme.net/sander/api/httpd/httpd_8h-source.html

הרעיון הוא לשחק עם SERVER_BASEVERSION (הפונקציה שמקבלת את הבאנר לפני ההצגה) או לפני בעזרת משחק עם ap_get_server_banner. לפרטים אפשר לפנות לכאן:

http://nohn.net/blog/view/id/removing_apache_server_header

פתרון נוסף הוא לפתוח את קובץ ה-httpd בעזרת Hex Editor ולערוך את השינויים על הקובץ המקומפל מבלי הצורך לקמפל אחד נוסף.

הורדת ה-X-Powered-By

כמו שראינו, בעזרת ה-X-Powered-By אפשר לראות איזו גירסאת PHP השרת מריץ. בכדי להסתיר אותה כך שלא תופיע בכל Response יש לגשת לקובץ ה-httpd.conf, ולהוסיף בשורה התחתונה:

```
Header unset "X-Powered-By"
```

במידה ולא איפשרתם את ה-mod_headers בסעיף הקודם יש לעשות זאת לפני כן. פתרון נוסף הוא להכנס לקובץ ה-php.ini ולקבוע:

```
expose_php = Off
```

בדיקה:

```
HTTP/1.1 200 OK
Date: Fri, 27 Nov 2009 00:33:56 GMT
Content-Length: 0
Content-Type: text/html
```

תוכלו לראות כי לאחר מהלך זה אין זכר לגרסאת ה-PHP.

ביטול התמיכה ב-HTTP OPTIONS/TRACE Methods

בכדי לקבוע באילו מתודות השרת יתמוך באופן קבוע, יש לטעון את המוד mod_authz_host (או mod_access בגירסאות החדשות) על ידי הורדת הסולמית בתחילת השורה:

```
#LoadModule authz_host_module modules/mod_authz_host.so
```

ולאחר מכן הוספת השורה הבאה בסוף הקובץ :

```
TraceEnable off
```

השורה הזאת תגרום לשרת להגיב ל-Trace Request עם 405:

```
HTTP/1.1 405 Method Not Allowed
Date: Fri, 27 Nov 2009 01:05:01 GMT
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Content-Type: text/html; charset=iso-8859-1
Content-Language: en
Content-Length: 961
```

במידה ונרצה לחסום את כלל המתודות על השרת חוץ מ-GET ו-POST, נוכל ליצור קובץ htaccess. על תיקית השורש שתכיל את הקוד הבא (יש לאפשר את ה-rewrite_module לפני השימוש בקוד):

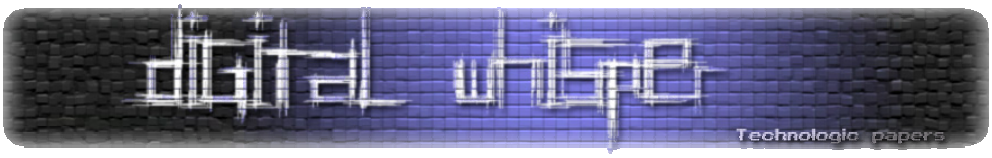
```
<LimitExcept GET POST>
deny from all
</LimitExcept>
```

נוכל לבדוק את הפתרון הנ"ל על ידי שליחת HTTP OPTIONS REQUEST:

```
OPTIONS /index.php HTTP/1.1
Host: localhost
```

ואכן נראה שאנחנו מקבלים 403 כתגובה, כמו במקרה הזה:

```
HTTP/1.1 403 Forbidden
Date: Fri, 27 Nov 2009 01:35:24 GMT
Vary: accept-language,accept-charset
Accept-Ranges: bytes
Content-Type: text/html; charset=iso-8859-1
Content-Language: en
Content-Length: 1096
```



לסיכום

הטכניקות שנגענו בהן במאמר זה לא יגרמו להפרעה לפעילות השוטפת של השרתים, אך יקשו על התוקפים הפוטנציאלים ולכן הן מומלצות. אם אתם מעוניינים לגרום לשרת שלכם באמת להיות "אנונימי" תוכלו לקחת בחשבון רעיונות נוספים כגון:

- שינוי סיומות קבצי ה-PHP/.NET לסיומות לא מוכרות- או סיומות הפוכות, החלפת PHP ב-ASPX למשל.
- ביטול השימוש האוטומטי ב-PHPSESSION/ASPSESSION והכנסת הפרמטרים הללו לקבצי ה-Cookies תחת שמות אחרים.
- קביעת דפי שגיאה נפוצים כגון 404 לדפי שגיאה גנריים של שרתים אחרים- כגון החלפת שגיאת ה-404 של שרת Apache בעמוד שגיאה של NET Framework.
- החלפת כלל הבאנרים המקוריים בסט באנרים של שרת אמיתי אחר- שרת ה-IIS יציג באנרים ותמיכה במודולים של שרת Apache.
- הדמיית תיקיות מודולי vti-bin/CGI בשרתים הפוכים.
- שינוי התצורה שבה מופיע ה-Time/Date ב-Headers לתצורה שונה.

אני בטוח שתוכלו לחשוב על עוד רעיונות. הדבר החשוב ביותר לזכור, כאמור, הוא שביטול הצגת ה-Fingerprints לא יגרום לשרת להיות מאובטח יותר, אלא יקשה על התוקפים לזהות את השרת שמולו הם עובדים ולכן פתרון זה צריך להיות חלק מאבטחת השרת, ולא כל אבטחת השרת.