



Web Application Security: Methods and Demos of Web Application Hacks

Presented by Paul Craig

Auckland

Brightstar - 12th Annual Security Summit

- Why Web Applications ?
- Know Your Enemy: Hacker Motives.
- Vulnerabilities and Exploits.
- Detecting a Vulnerability.
- Tools of the Trade.
- Web Application Hacking Examples.
- Getting Secure, and Staying Secure.
- Conclusion.

Question:

I know IT staff have a busy life

IT security is a huge subject and web applications make up a very small part of IT security.

But why are web applications the most important part of security, and potentially the biggest weak spot in your entire network.

Do you have an answer?

- I pondered this question for some time.
- The answer: Firewalls
 - `iptables -A INPUT -p tcp -s 0/0 -d 0/0 --destination-port 80 -j ACCEPT`
 - `iptables -A INPUT -p tcp -s 0/0 -d 0/0 --destination-port 443 -j ACCEPT`
 - `iptables -A INPUT -s 0/0 -d 0/0 -p tcp -j DROP`
- More Simply:
 - Web applications are the only thing we still allow into our DMZ.
- Everyone allows port 80/443 (from anywhere) to a web server.

- We are all 'security minded people'
 - Your servers are patched to the latest patch level.
 - Host firewalls enabled.
 - Employee's use a 10 character secure password.
 - You only hire skilled security professionals.
 - You attend security conferences!
- I asked a windows admin friend of mine
"Why are your IIS web servers secure? How do you know?"
- He Replied
"I use a 'Secure Windows Server' deployment checklist.
Basically, I make sure every checkbox is ticked. "

- Security thrives on policy and checkboxes.
 - Windows Firewall Enabled
 - Auto Update Enabled
 - Anti Virus Installed
 - Secure User Passwords
 - Host Hardening Procedure
- These are things you can visually sight and tick off.
- You can verify that Windows is secure, and it probably is.
- But what about your web application?
- You ASSUME its secure, you hired good developers right?
- How good are your C# or JSP skills?

It's a pity web applications do not come with an configuration option

Secure Mode



- In other words.
- The only thing we allow into our DMZ, is the only thing we cannot verify the security of.
- If a developer came to you and asked.
 - “Can I get an external firewall rule to allow internet access to this cool service I run on my desktop”
- You would (hopefully) say.
 - “NO!, an exposed service no one knows anything about is a security risk.”
- Would you even know if the same developer uploaded a single page web application to your website?
 - test.asp, debug.aspx, test.jsp etc.
- There is no difference between an ‘unknown service’ and a web application, its all code your server will execute.
 - How many of you know about every file in your webroot?
- Code written by developers, who make mistakes.

- How has security changed in the last 5 years? (2002)
 - 5 years ago firewalls were not widely deployed and often poorly configured.
 - *Personal Firewalls* or *Desktop Firewalls* were unheard of.
 - Patching was not seen as a high priority task, worms were only just gaining popularity (CodeRed launched 2001)
 - Many, many un-patched NT servers on the internet which have never been compromised.
- Hackers 5 years ago focused on exploiting vulnerabilities within network based protocols (RPC/SMB/MS-SQL/IIS)
 - Network protocols usually ran as a privileged user.
 - Protocols are easily accessible due to the lack of firewalls.
- There were 82,094 security incidents reported in 2002. (cert.org)

- Then came the worms!
 - *CodeRed, CodeRed II, Slammer, Bagle, Blaster, Netsky, SoBig, MyDoom, MyTob.*
 - Media frenzy, "Worms are the greatest threat of the internet."
- EVERYONE installs at least one firewall!
- HUGE boom in security related software products.
 - Estimated in 2004 the security software industry was worth \$7.7 billion (USD)! (over 200% gain from 2003)
- In the present day, everyone uses two firewalls, three Anti-Virus products and prays to four different deities for anti-hacker protection.
 - Firewalls are seen as the golden bullet, deny the traffic and keep secure.
- Today, an un-patched NT server will last under 5 minutes on the internet before a worm compromises it.
 - We are becoming more paranoid, more worried.

- By installing so many firewalls we have forced hackers to evolve, to change how, and what they attack.
- The new target is the only thing we still allow into our DMZ's.
 - HTTP/HTTPS. (Port 80/443)
 - Web Applications!
- Hackers of today exploit web applications, not network services.
- Web applications are typically easier to exploit than network services.
 - There are billions of active web applications on the internet, one web server may host many different web applications.
 - Kids can, and do hack web applications.
 - Remote command execution in seconds.
 - Even on a fully patched up-to-date Windows server.
- Our attempt at making the internet secure has failed.
 - Firewalls are not the golden bullet.



Motives for Hackers

Top #3 Malicious Motives:

- **Identity Theft**
- **Hacktivism**
- **Defacement**

#1 – Defacement

- New age digital graffiti artists.
- Hackers seek fame, not fortune, or a political agenda.
- Hackers work in groups, trying to make their group or country listed top on defacement archive websites.
 - Turkish hacker “iSKORPiTX” is currently ranked #1 on zone-h.org (A popular defacement archive portal).
 - The very same hacker hacked x2.quik.com (Vodafone -IHUG owned US hosted server) in February this year.
- According to zone-h.org.
 - Total defacements in .NZ: 4,415 of which 950 single IP and 3,465 mass defacements

HACKED BY iSKORPiTX

TURKISH HACKER

dünya markasi taklit edilemez

#2 - Identity Theft.

- “Federal Trade Commission (FTC) claims 42% of identity theft cases involved credit card fraud from stolen credit card details”.
- My own credit card details are currently stored in 10-15 different databases around the world.
- My personal credit card is worth \$20-\$50 (USD) on the black market.
- Your company credit card is worth \$50-\$100.
- Hackers steal credit cards and on-sell them to *carders* who order products online to resell on online auction sites.
- Money is a good incentive.
- Rise in organized crime involvement in carding.
 - 2006, 50 hackers from Egypt and Lebanon were arrested for stealing millions of dollars from stolen credit and debit card numbers.
 - Hackers were found to be working for middle eastern extremist groups. (Al-Qaeda & Hezbollah)

#3 - Hacktivism.

“Hacking with the intent of political or social change.”

- Protesting without having to leave the house.
- The Iraq War
 - March 2003, 20,000+ American sites defaced in one week with anti-American/anti-war slogans.
- USA Sells arms to Taiwan, China retaliates.
 - Hundreds of US governmental sites defaced to protest sale of arms to Taiwan.
- Islamic Hackers retaliate over Danish Muhammad cartoons.
 - Over 1,000 Danish websites defaced by Islamic hackers protesting infamous ‘Muhammad cartoons’

- A term your going to hear a lot, “Web Application Vulnerability”
- What does it really mean?
 - A vulnerability exists when a remote user can create an unexpected situation.
 - The vulnerability is allowing a user to do something ‘unexpected’, something the application was not originally designed to do.
 - Your using the web application creatively, but potentially not maliciously.
- What is an Exploit?
 - An exploit exists when an unexpected event can be leveraged to gain control.
 - The exploit is using the unexpected event to your advantage and taking control of the server or application.

- Example:
 - The page edituser.asp accepts an integer "UserID" value as input.
 - `http://www.host.com/Edituser.asp?UserID=112`
 - The UserID of user Paul is 112.
 - If you modify the UserID to be 113, Paul can edit Bob's users account.

- The web application was not expecting you to change the UserID value.
- By supplying the UserID of another valid user, the application is exploited.

- Ok, so not all vulnerabilities are so obvious.
- Finding vulnerabilities is serious brain crunching work.
 - After eight hours of auditing I can barely think straight.
- Your finding a situation the developer did not plan for.
 - Its like chess, your out thinking the developer.
- Not all developers are created equal.
 - There are always flaws!
 - Only 1-2% of applications reviewed are flawless.
- Its about getting creative.
- Trying something different, something strange the developer didn't think of when the application was written.

- Good security testers tend to have a very twisted, bent mind.

- How do you find a vulnerability?
- There is one rule I use when auditing web applications.
 - Keep your eyes open, pay very close attention!
- As a penetration tester you are trying to find unexpected functionality.
- Unexpected functionality OFTEN results in an error, or something 'different' happening.
 - Page does not load, error message created.
 - Session dies, cookie changes.
 - Web browser incorrectly renders HTML contents.
 - Pages takes 3 seconds longer to load.
- All minor details are incredibly important!
- Pay attention to everything.

- Three Easy Steps To Detect A Vulnerability
- #1 – Determine WHAT do you control.
 - Test.asp?readfile=<USER INPUT>
 - User input is directly passed to a 'read file' function.
 - We control the filename argument passed to an fopen call (File Open)
- #2 – HOW can we use the functionality that we control?
 - After the file is opened, fread() is used to display the contents of the user supplied filename.
 - Try to open
 - ../../secrets.txt, Test.asp, \\device\harddisk

- #3 – Take Control:
 - Use any influence over the web application to leverage access.
 - Control over an fread() function allows a malicious user to.
 - View the contents of sensitive IIS files.
 - Global.asa, web.config.
 - Both these files can contain database and domain user credentials.
 - Review the source of other script files for security flaws.
 - Try to access other sensitive files

- Hollywood makes hacking out to be visually amazing.
- 3 dimensional worms, computerized voices, big red lights.
- Sadly this is not the case! ☹

- Real web application 'hacker tools' are very simple and rather uninteresting.
- Sorry to disappoint.
 - Web application hacking tools focus on one main objective, Input Manipulation.

- Manipulate all aspects of input sent from the client browser to the web application.

- The most important tool I use.
- Web Proxy:
 - Designed to sit between the local web browser and the remote web server.
 - “Man In The Middle” attack on yourself.
 - Provides access to monitor and modify all requests sent to, and from the web application.
 - Cookie variables, HTTP POST, HTTP GET, Extra headers.
 - Allows us to change variables at will, and see what happens to the application.
 - Similar to an application debugger, only for web applications.

Intercept requests : Intercept responses :

Parsed Raw

Method **URL** **Version**
 POST https://login.live.com:443/ppsecure/post.srf?id=2&svc=mail&cbid=24325&msspjph=1&tw=900&fs=1&lc=2057&_lang=EN&bk=11736824 HTTP/1.1

Header	Value	
Host	login.live.c...	▲ ▼
User-Agent	Mozilla/5.0 ...	

Insert

Delete

URLEncoded Text Hex

Variable	Value	
PPSX	Passpor	Insert Delete
PwdPad	IfYouAreReadingThisYouHaveTooMuchFree	
login	test@test.com	
passwd	test	
HIPChallenge	4bRbID04hCDE0sVkBgCJCf0dILC9sYU4Eswk5kPc9theiZVnTTs1f8Tr...	
HIPSolution	CJYHCX	
PPFT	B4KfcOG56yqeQV0w4QC0h4gkAgDDTKpjVW03TiUiaue28I4MGFdje4Vi...	

Parsed Raw

Version **Status** **Message**

Header	Value

Hex

Position	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	String

Accept changes

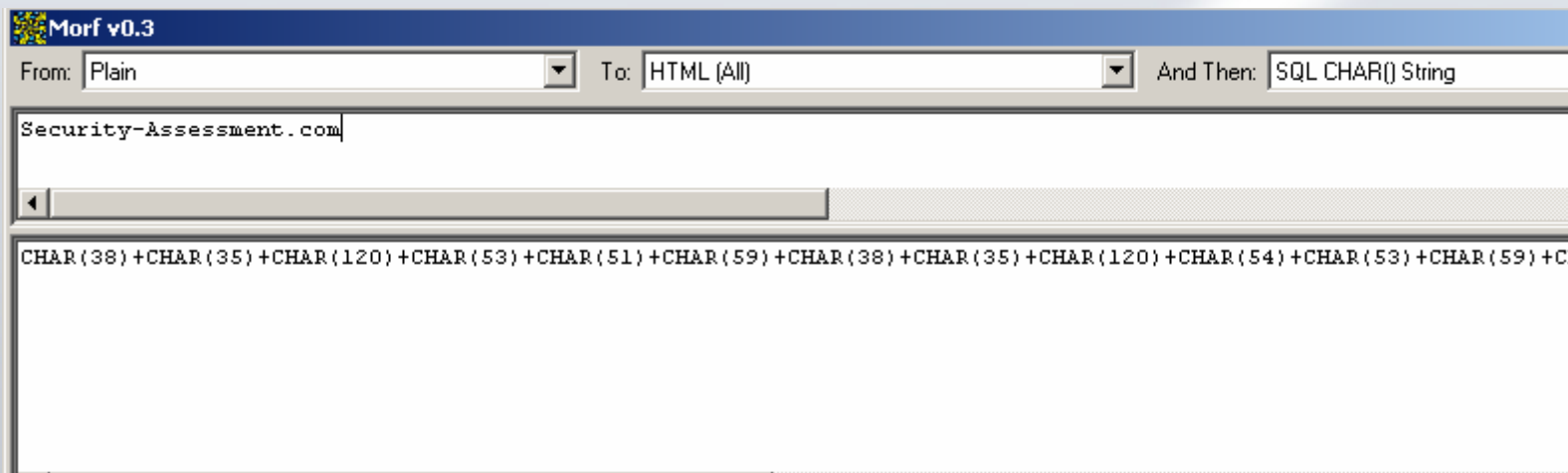
Cancel changes

Abort request

Cancel ALL intercepts

- Text Encoding Utilities.
 - Text encoding techniques are often used as a method of bypassing security checks.
 - Creating an unusual situation, by changing how the data you supply is encoded.
 - Example: test.cgi prohibits ../ from any user supplied filename, but does not deny %2e%2e%2f (../ in encoded in Hex)
 - Web Server decodes the Hex encoded ../, test.cgi does not.
 - Infamous IIS4/5 Unicode directory traversal exploit.
 - Use an alternate extended Unicode encoded / character to perform a directory traversal attack and access cmd.exe.
 - /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
 - /../ is filtered, but `/\..%c0%af` is not!
 - Technically it's the same as /, it just looks completely different.

- IO:Active Morf
 - www.ioactive.com
 - “Morf is the supreme ninja god of encodings. ”



- Allows double encoding: Plaintext>URL>SQL Char
- Supports: Url,Html,base64,Hex,ViewState, PEM, UTF-7,...
- Basically we can encode any text and encode it into any other encoding format.

- Automated Web Application Vulnerability Scanners.
 - Otherwise known as *Fuzzers*.
 - Software testing method first developed in 1989.
 - *Fuzzers* send random data *Fuzz* until 'something different' happens.
 - I.E, an error message is produced.
 - An unexpected event occurs.
 - *Fuzzers* are good at finding obvious vulnerabilities but are unable to find complicated vulnerabilities.
 - Recent tests performed by Security-Assessment.com found commercial web application fuzzing software were only able to detect 40% of vulnerabilities!
 - Not even half!
 - *Fuzzers* are good at doing grunt work; finding the really obvious vulnerabilities.

Enough talk, lets hack something.

- Example #1 – Dog Food Factory Press Releases
 - A PHP developed script which displays company press releases.
 - Press releases are kept as .txt files on the web server.
 - A user supplied filename variable is used to specify which press release to include.

- `/release.php?release=press/release1.txt`

- Example #2 – ‘Share Your Holiday Snaps’ ASP Image Upload
 - An ASP developed page for employee’s to upload photos of their holiday adventures.
 - Images are stored on the web server for everyone to see.
 - The upload script was developed to ONLY accept files with an extension of .JPG.
 - If the last four characters of the filename are not .JPG, the file is rejected.
 - Our goal is of cause to upload malicious .ASP files.

- Example #2 - Analysis of the vulnerability
 - test.asp is rejected.
 - test.jpg is accepted.
 - ASP compares the last four characters of the filename to “.JPG”
 - Our filename of test.asp<NULL>.JPG is accepted, but written to the disk as test.asp, WHY?
 - According to ASP the filename string ends with .JPG
 - However, when writing the uploaded file to disk underlying Win32 API calls terminate a filename string at the first NULL byte found.
 - Windows thinks test.asp<NULL>.jpg = Test.asp
 - ASP thinks Test.asp<NULL>.jpg = Test.asp<NULL>.jpg
 - ASP considers NULL bytes as just data, the entire string is seen.
 - Windows treats a NULL byte as the end of a string!

- Example #3 – My Goldfish Portal
 - A place for like minded people to share goldfish tips and tricks.
 - Users are authenticated from a backend SQL database.
 - Only legitimate, authenticated users are given access to the Goldfish resources.
 - We need to gain a valid user account.
 - Compromise the server.

- Example #3 - Analysis of the vulnerability
 - SQL Injection allows a malicious user to control a SQL query.
 - Queries can be injected to enumerate databases, tables and fields.
 - SQL error messages can be used to return data from the database.
 - “cannot convert string Mysecretpassword to integer”
 - SQL injection is the most common form of command execution and the most common class of vulnerability we find.
 - Appending -- sp_password will cause SQL server to not log the query in any transaction logs
 - “This query contains sensitive information”

- Example #4 – Google Hacking and Google Dorks.
 - Google hacking was made famous by Johnny Long's book "Google Hacking" and his 2005 presentation at BlackHat/Defcon
 - Using Google to find vulnerable web applications.
 - The script kiddies of yesterday have become the Google dorks of today.
 - One difference, Google hacking is VERY easy.
 - When a web application security advisory is released, the advisory often contains a 'Dork' and an attached exploit.
 - The 'Dork' is the Google search term to use to find instances of the script.

"powered by WebText" - Google Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.google.co.nz/search?q=%22powered+

Getting Started Latest Headlines

[Web](#)
[Images](#)
[Groups](#)
[News](#)
[more »](#)

[Advanced S](#)
[Preferences](#)

Search: the web pages from New Zealand

Web Results **61 - 70** of about **9,570** for **"powered by WebText"**

[WebText 0.4.X Evolution :: Przykładowa strona](#)
 Użytkowników: , 1. On-line: , 1. Odwiedziny: , 61. Logowanie. Login: Hasło:. Copyright © by
 Twoje dane All rights reserved. **Powered by WebText** v0.4.5 Evolution.
[mytibia.fasthost.pl/?sekcja=ksiega - 7k - Cached - Similar pages](#)

[Izodek - Web Design](#)
 All rights reserved. **Powered by WebText** v0.4.5 Evolution Valid XHTML 1.0 Transitional ·
 Valid CSS! Pobierz Firefoks! Nowoczesne przeglądarki.
[izodek.maghost.net/ - 6k - Cached - Similar pages](#)

[MAREK SIWEK](#)
 1754. Autor tekstów, wokalista, recitale. Koncert "Symbioza" z grupą Lombard, płyta "Białe
 sale". Copyright © by Mantis Design **Powered by WebText**.
[www.mareksiwek.pl/ - 5k - Cached - Similar pages](#)

[MAREK SIWEK](#)
 1737. GALERIA. Symbioza. Jesteśmy Razem. Recital. Imprezy Charytatywne. Copyright ©
 by Mantis Design **Powered by WebText**.
[www.mareksiwek.pl/?go=galeria&PHPSESSID=2c88f3a8cb1e46301ce941509afb346 - 8k -
 Cached - Similar pages](#)
[\[More results from www.mareksiwek.pl \]](#)

[NSZZ "Solidarność '80" KWK "Jas - Mos"](#)
 Copyright © by NSZZ Solidarność '80 / Designed, CMS Modified & Gallery by www.UML.pl /
Powered by WebText v0.4.5 Evolution | [admin] [ftp]
[www.s80jm.nazwa.pl/ - 6k - 14 Mar 2007 - Cached - Similar pages](#)

- 9,570 hits
- Popular in Poland
- Its just too easy.

Getting Secure, and Staying Secure

- Breaking web applications is not hard.
- All it takes is a single development mistake, one bug.
- Getting secure and staying that way requires awareness.
- Ensure your business correctly understands the risks associated with web applications.

- Web application security is as important as application functionality or user interface design.
- Security must be implemented from conception!
 - It saves you time and money in the long run.
- Developers must understand that security is a requirement!

- Top security recommendations from a security auditor.
 - Trust me, I write recommendations for a living.

- Be Aware Of Your Document Root.
 - Ensure the document root ONLY contains content you know of.
 - Remove all other scripts, test scripts, backups, samples, etc.
 - Reduce the attack surface.

- Avoid using free or commercial pre-made web applications on critical web servers.
 - Example: Forums, chat, image gallery applications, CMS.
 - Single point of failure.
 - Google dorks!

- Log Files.
 - Web server log files are vital!
 - Log as much data as possible.
 - Extra HTTP headers, X-FORWARDED-FOR
 - Archive log files.

 - Use log analysis tools
 - Example, WebTrends
 - Watch for trends and anomalies.
- Remember: If you do get compromised, log files are the first thing an external forensic auditor will ask for.

- In-House Auditing
 - Implement an internal peer review system between developers.
 - Reward employee's who identify security flaws.
 - A \$20 Liquor Land voucher per vulnerability found is a cheap, and highly effective insensitive.
 - Many security flaws are easy to spot!
 - Simple peer review may find 40-50% of the vulnerabilities present.

- Call in the auditors!.
 - Developers are not security experts
 - In-house auditing is the first line of defence.
 - Internally you can find many vulnerabilities, but not everything!
 - Bring in the auditors as early as possible!
- Security reports allow developers to identify their own flaws.
- Most developers don't make the same mistake twice.
- Third or fourth consecutive web application review is always MUCH harder than the first.
- Developers learn how to write secure software.

- Security Education
 - Buy your developers books on secure software development.
 - Send your developers to security conferences.
 - Ensure developers understand the importance of application security.
 - Security should NEVER be tacked on the end of a project.
 - Project managers must understand security is INTEGRAL, it should be interwoven into the project from conception, not in the last 5%!
 - Oh S%\$!T, need a security review, we go live tomorrow!
- Keep doing everything else.
 - You still need firewalls, both host and perimeter.
 - Keep servers up-to-date with patches.
 - Up to date anti virus.
 - Tick every checkbox.

- HTTP/HTTPS is the only protocol still allowed into most networks.
- Web application hacking begins with finding an unexpected situation.
- Use the unexpected situation to gain control and exploit the application.
- It only takes a single line of insecure web application code to compromise a server.

- Raise awareness regarding web application security.
- Security **MUST** become part of the application development lifecycle.
- Implement security from conception!

Questions?

Comments?

paul.craig@security-assessment.com