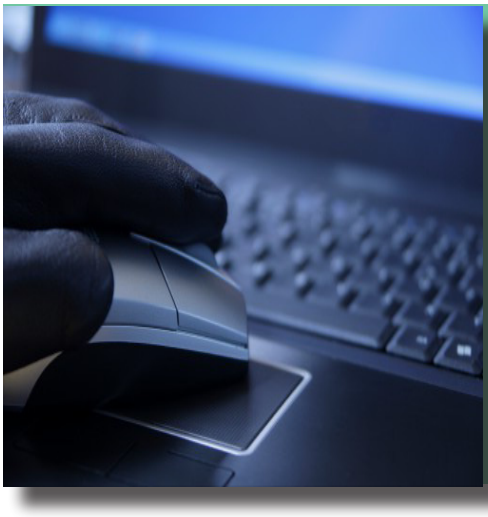
BREACH



TOP WEB INCIDENTS AND TRENDS OF 2009 AND PREDICTIONS FOR 2010

BY: RYAN BARNETT

JANUARY 2010

Breach Security, Inc. | Corporate Headquarters | 2141 Palomar Airport Road, #200 | Carlsbad, CA 92011 | USA
tel: (760) 268-1924 | toll-free: (866) 205-7032 | fax: (760) 454-1746 |www.breach.com

OVERVIEW

The purpose of this white paper is to present a high level view of web application security incidents, trends and predictions so that they may raise awareness of the types of vulnerabilities that are being targeted by cyber criminals.

This white paper includes data from the web hacking incident database (WHID) (http://www.xiom.com/whid-about) which is a project dedicated to maintaining a list of web application-related security incidents. The WHID's purpose is to serve as a tool for raising awareness of the web application security problem and provide information for statistical analysis of web application security incidents. Unlike other resources covering website security, which focus on the technical aspect of the incident, the WHID focuses on the impact of the attack. To be included in WHID an incident must be publicly reported, be associated with web application security vulnerabilities and have an identified outcome. Breach Security Labs (http://www.breach.com/resources/breach-security-labs/) is a WHID project contributor. For further information about the Web Hacking Incidents Database refer to http://www.xiom.com/whid-about.

TOP 10 WEB INCIDENTS OF 2009

The purpose of this section is to highlight a few of the top interesting web incidents of the past year. The idea is to present different types of attacks and outcomes and to offer some lessons learned.

TJX/HANNAFORD/HEARTLAND HACKERS CAPTURED

New information from the capture and trial of the identity theft ring leader Albert Gonzalez (http://www.time.com/time/business/article/0,8599,1917345,00.html), reveals that in order to penetrate TJX, Hannaford and Heartland networks from the captured end points, the hackers employed different techniques including password sniffing and SQL injection. The later justifies getting the TJX incident for the 1st time into WHID. The following data was presented by a joint FBI/Secret Service Advisory (http://www.ic3.gov/media/2008/081215.aspx) outlining the common methodologies used in these attacks -

1. *They identify Web sites that are vulnerable to SQL injection. They appear to target MSSQL only.*
2. *They use "xp_cmdshell", an extended procedure installed by default on MSSQL, to download their hacker tools to the compromised MSSQL server.*
3. *They obtain valid Windows credentials by using fgdump or a similar tool.*
4. *They install network "sniffers" to identify card data and systems involved in processing credit card transactions.*
5. *They install backdoors that "beacon" periodically to their command and control servers, allowing surreptitious access to the compromised networks.*
6. *They target databases, Hardware Security Modules (HSMs), and processing applications in an effort to obtain credit card data or brute-force ATM PINs.*
7. *They use WinRAR to compress the information they pilfer from the compromised networks.*

Lessons Learned

PCI requirements focus on "externally facing web applications" and unfortunately many organizations end up forgetting to properly security internal web applications to the same level. As outlined in these new reports, once the hackers were able to get on an unsecured wireless access point, they were able to launch SQL Injection attacks against internal systems.

References

WHID 2007-89: The Big TJX Hack - http://www.xiom.com/whid-2007-89
WHID 2008-52: The Hannaford Breach - http://www.xiom.com/whid-2008-52
Heartland Data Breach at Datalossdb - http://datalossdb.org/incidents/1518-malicious-software-hack-compromises-unknown-number-of-credit-cards-at-fifth-largest-credit-card-processor
WHID 2009-29: FBI & Secret Service Warn of a Sophisticated HMS Attack - http://www.xiom.com/whid/2009/29/HSM_Attack

TIME'S MOST INFLUENTIAL POLL ABUSE

Polls are easy target for automation abuse. You can usually participate anonymously and the poll operator has an interest in drawing as many participants as possible, but as demonstrated by previous incidents such loose security enables hackers to distort the results.

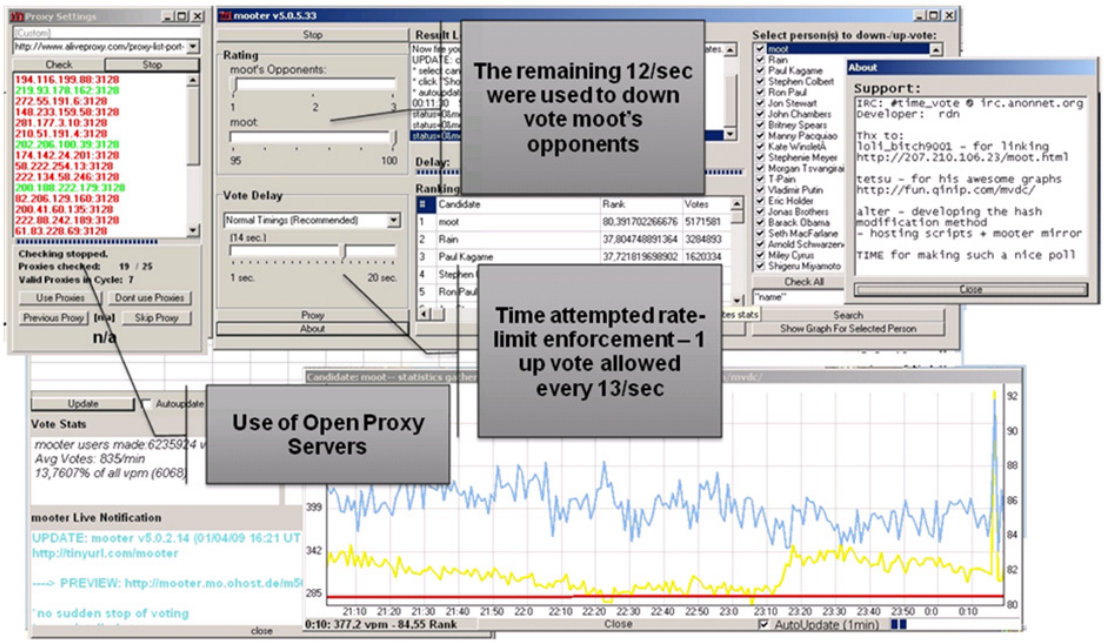
This time a hacker succeeded in manipulating Time's poll for most influential people in 2009. Such polls are probably always distorted by automated programs, with every stakeholder running his own robot to promote a cause. The time poll status shown above includes mostly known people, though the standings do seem skewed. Is it just that our view of the world is different than others, or have Muslims around the world become avid Time readers? The top rated person, "moot", which none of you heard about until now, proves that it is all about automation.

This specific poll distortion reported by Paul Lamere is unique since a group of hackers called 4chan, led by "moot", took the time to fight Time's humble attempts to mitigate automation. Among the measures and countermeasures that 4chan and Time exchanged are:

- 4chan distributed the simple get URL required to vote for moot through legitimate web sites and comment spamming. Such a link can easily be executed automatically by a web site user without his awareness using CSRF techniques. Auto-voter SPAM link URL:
    - http://fun.qinip.com/gen.php?id=1883924&rating=1&amount=200
- Using a typical CSRF counter measure, Time added a salted and hashed key to ensure that the poll was submitted from its own poll form. However the key was authentication on the client by Time's poll Flash application enabling 4chan to easily find it out and overcome the issue. Once the key was known, 4chan was able to include it within more SPAM/CSRF links:

```
<html>
<head>
<title>
</title>
</head>
<body>
<img src="http://www.timepolls.com/hppolls/votejson.do?callback=processPoll&id=335&choic
e=1&key=a4f7d95082b03e99586729c5de257e7b" />
<imgsrc="http://www.timepolls.com/hppolls/votejson.do?callback=processPoll&id=335&choice
=1&key=a4f7d95082b03e99586729c5de257e7b" />
...
</body>
</html>
```

- The Time voting mechanism did not even check that the ranking in the vote was legal, so a link to vote down "moot" competitors in the list was also used until Time fixed the issue. Voting down is key to winning such a poll as 4chan competitors are not at rest running their own sophisticated campaigns.
- Lastly 4chan developed sophisticated robots to auto-vote. Those robots overcome Time's anti-automation protections: since each user is allowed to vote just once in every 13 seconds, the robots uses open proxies to vote faster. Since time only prevents voting for the same person from the same IP, the robots used the extra 12 seconds available for each source IP to vote down competitors. The system also reports to a central server allowing monitoring of the voting rate!



However this specific hack is ever more interesting. At one point 4chan were bored with just running moot for presidency, so they decided to use their sophisticated machine to do a more elaborate work. They decided to fix all first 21 nominees so that their initials would spell "Marblecake Also the Game". And as Paul Lamere's screenshot proves, they made it.

Rank	Name	Avg. Rating	Total Vote
1	moot	87	12,939,521
2	Anwar Ibrahim	42	1,632,411
3	Rick Warren	42	1,290,988
4	Baitullah Mehsud	40	1,281,854
5	Larry Brilliant	39	1,425,061
6	Eric Holder	38	1,215,008
7	Carlos Slim	37	1,311,525
8	Angela Merkel	37	1,069,787
9	Kobe Bryant	36	1,195,005
10	Evo Morales	34	1,045,245
11	Alexander Lebedev	34	640,115
12	Lil' Wayne	33	637,426
13	Sheikh Ahmed bin Zayed Al Nahyan	32	622,054
14	Odell Barnes	31	621,182
15	Tina Fey	30	646,446
16	Hu Jintao	29	614,359
17	Eric Cantor	28	580,189
18	Gamal Mubarak	27	580,389
19	Ali al-Naimi	26	627,786
20	Muqtada al-Sadr	25	564,094
21	Elizabeth Warren	24	559,800
22	Manny Pacquiao	23	9,382,234

Lessons Learned

Insufficient Anti-Automation defense, which is the ability to identify and react to non-human clients, is a critical issue for most web applications. Failure to do this can lead to successful Denial of Service, Brute Force and Scraping types of attacks.

References

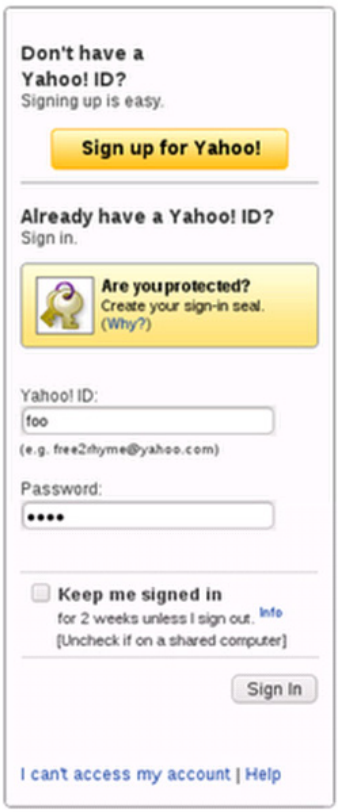
The Register – Hackers stuff ballot box for Time Magazine's top 100 poll (http://www.theregister.co.uk/2009/04/17/time_top_100_hack/)
Inside the precision hack (http://musicmachinery.com/2009/04/15/inside-the-precision-hack/)
WASC-21: Insufficient Anti-Automation (http://webappsec.pbworks.com/Insufficient+Anti-automation)

BRUTE FORCE ATTACKS AGAINST YAHOO WEB SERVICES

As part of the WASC Distributed Open Proxy Honeypot Project (DOPHP) (http://projects.webappsec.org/Distributed-Open-Proxy-Honeypots), we have been able to track some pretty extensive distributed brute force attacks against Yahoo end-user email accounts. Valid email accounts and/or obtaining valid account credentials are a huge commodity for SPAMMERS. Identifying valid accounts is important as it allows them to only send SPAM messages to real accounts and they can also be able to sell lists of valid accounts to other SPAMMERS. Taking this a step further, if the SPAMMERS are able to enumerate valid credentials for an account (username and password) they can then hijack the account and use it for SPAMMING.

Normal Web Login

This methodology is not new and Yahoo is obviously aware of these attacks aim at their Yahoo mail web login interface page (https://login.yahoo.com/). This login page looks like this -



When a client clicks submit, the request looks similar to the following -

```
POST /config/login? HTTP/1.1
Host: login.yahoo.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: https://login.yahoo.com/
Cookie: B=ffetg09557ar5&b=3&s=od; cna=zwISA2sCdzgBAS+RbUtyRRes; Y=%2e
Content-Type: application/x-www-form-urlencoded
Content-Length: 296

.tries=1&.src=&.md5=&.hash=&.js=&.last=&promo=&.intl=us&.bypass=&.partner=&.u=007ofj55asupi&.v=0&
.challenge=hKhk9.OX5y0EOqJ3c4yxAH _ rSrx5&.yplus=&.emailCode=&pkg=&stepid=&.ev=&hasMsgr=0&.
chkP=Y&.done=http%3A%2F%2Fmy.yahoo.com&.pd= _ ver%3D0%26c%3D%26ivt%3D%26sg%3D&login=foo&passwd=b
ar&.save=Sign+In
```

Notice the in the post payload that the application is tracking how many “tries” have been attempted. This is useful for throttling automated attacks and once a client goes over a limit, Yahoo then presents the user with an added CAPTCHA challenge -

Also notice that the login page is presenting the end user with a generic error message indicating that the credentials were not correct but it does not inform the user whether it was the login or password that was wrong. All of this type of anti-automation defense is good. The problem is - is Yahoo applying this type of defense consistently throughout their entire infrastructure? Are there any ways for the SPAMMERS to find a backdoor? Unfortunately, yes.

Web Services APP

"Security is lax on this side."

The WASC DOPHP has identified a large scale distributed brute force attack against what seems to be a web services authentication systems aimed at ISP or partner web applications. The authentication application is named "/config/isp_verify_user". Google links for the "isp_verify_user" app are here (http://www.google.com/search?q=inurl:/config/isp_verify_user&hl=en&client=firefox-a&rls=org.mozilla:en-US:official&filter=0). One thing you will notice in looking at these results is that there is an incredibly large number of Yahoo authentication subdomains that are hosting this application and are able to authenticate clients. If you click on one of the links, you will see that the response data returned in the browser is terse. It is simply 1 line of data such as this -

```
ERROR:210:Required fields missing (expected l,p)
```

The format of this data is obviously not intended for end users, but it more tailored for parsing by web service applications. It very well could be that many front-end web applications are validating the credentials submitted by clients to these isp_verify_user app. This particular error message is returned when a client does not submit the l (login) and p (password) parameters. If a client sends a request for a login/username that does not exist (http://www.google.com/search?q=inurl:/config/isp_verify_user+ERROR:102:&hl=en&client=firefox-a&rls=org.mozilla:en-US:official&filter=0), the app will return a message of -

```
ERROR:102:Invalid Login
```

Remember the generic error message presented on the normal login web page? Not here - it is easy for a SPAMMER to automate sending requests and cycling through various login names to identify if/when they hit on a valid Yahoo account name. When this happens, the application gives a different Invalid Password (http://www.google.com/search?q=inurl:/config/isp_verify_user+ERROR:101:&hl=en&client=firefox-a&rls=org.mozilla:en-US:official&filter=0) error message -

```
ERROR:101:Invalid Password
```

Note that this application does not implement any of the same CAPTCHA mechanisms that the standard login page does. This means that the attackers have an unimpeded avenue of testing login credentials. If the client sends the correct credentials (http://www.google.com/search?q=inurl:/config/isp_verify_user+OK:0:&hl=en&client=firefox-a&rls=org.mozilla:en-US:official&filter=0), they will receive a message similar to the following (where username is the data submitted in the "l" parameter) -

```
OK:0:username
```

With this information, the SPAMMERS can then log into the enumerated email account and abuse it as they wish.

Scanning Methodology

Here is a small snippet of some of the transactions that were captured -

Get
http://l33.login.scd.yahoo.com/config/isp_verify_user?l=kneeling@ort.rogers.com&p=qwerty HTTP/1.0
Get
http://l06.member.kr3.yahoo.com/config/isp_verify_user?l=kneading@ort.rogers.com&p=000000 HTTP/1.0
Get
http://69.147.112.199/config/isp_verify_user?l=kitbags@ort.rogers.com&p=333333 HTTP/1.0
Get
http://217.12.8.235/config/isp_verify_user?l=kirk@ort.rogers.com&p=yankees HTTP/1.0
GET
http://69.147.112.217/config/isp_verify_user?l=__miracle&p=weezer HTTP/1.0
GET
http://69.147.112.202/config/isp_verify_user?l=123#@!.._69_&p=weezer HTTP/1.0
GET
http://68.142.241.129/config/isp_verify_user?l=__lance_&p=weezer HTTP/1.0
GET
http://202.86.7.115/config/isp_verify_user?l=__kitty__69__&p=weezer HTTP/1.0

The attackers used a three dimensional scanning methodology as described below -

1. Distributing the scanning traffic through multiple open proxy systems. This changes the source IP address as seen by the target web application so basic tracking/throttling is more challenging.
2. Distributing the traffic across different Yahoo subdomains. The advantage to this is that even if some form of failed authentication tracking is taking place, it is more difficult to synchronize this data across all systems.
3. Diagonal scanning - submitting different username/password combinations on each request. This is instead of vertical scanning which is choosing 1 username and cycling through passwords or horizontal scanning which is choosing 1 common password and cycling through userenames.

Lessons Learned

1. Implement proper ACLs against all web services apps. In this case, the isp_verify_user app was clearly not intended for direct client usage however there are no ACLs that prevent an end user from accessing them.
2. Need to identify any rogue web application authentication interfaces. This is a big problem for organizations that are either newly deploying distributed web services apps or those who have newly acquired a business partner.
3. Every web application must have some form of anti-automation capability in order to identify when clients are sending these requests.

References

SC Magazine - Rampant Brute Force Attacks Against Yahoo (http://www.scmagazineus.com/rampant-brute-force-attack-against-yahoo-mail/article/149373/).
SearchSecurity – Brute Force Attacks Target Yahoo Email Accounts (http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1368227,00.html)

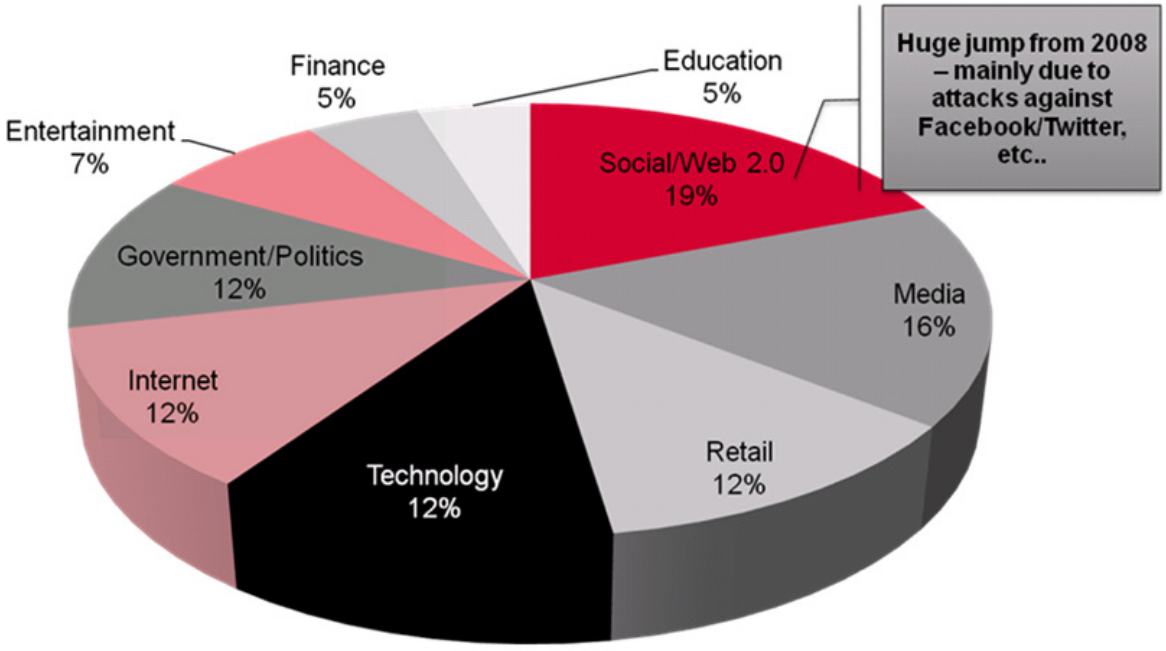
TOP WEB TRENDS OF 2009

WHID INCIDENTS BY ATTACKED VERTICAL

Do you remember that line from the movie Field of Dreams: "If you build it, they will come"? Well, according to the data captured from the Web Application Security Consortium (WASC) Web Hacking Incidents Database (WHID) project, online criminals are re-enforcing that movie quote. The fact is that profit driven criminals have learned that they can utilize social networking types of web sites (such as Twitter, Facebook and MySpace) as a means to target the huge number of end users.

Breach Security Labs, a WHID contributor, analyzed the WHID events from 2009 and it was found that Social Networking sites (such as Twitter) that utilize Web 2.0 types of dynamic, user-content driven data, are the #1 targeted vertical market. The reason for this is really two-fold:

1. Criminals are now directly targeting the web application end-users. The bad guys are using flaws within web applications to attempt to send malicious code to end users. Popular websites that have large user bases are now ripe targets for criminals. These are target rich environments.
2. Social networking sites are so popular partly because they allow their users to customize and update their accounts with user-driven content, widgets and add-ons. These features make the sites dynamic and fun for the end users, however they also unfortunately also significantly increase the cross-site scripting (XSS) and cross-site request forgery (CSRF) attack surfaces.



"The dramatic rise in attacks against social networking sites this year can primarily be attributed to attacks on popular new technologies like Twitter, where cross-site scripting and CSRF worms were unleashed," said Ryan Barnett, director of application security research for Breach Security.

Lessons Learned

Organizations that are considering using Web 2.0 types of technologies must conduct a thorough threat modeling exercise to identify all weaknesses. Specifically, these types of applications rely heavily upon user-driven dynamic content so it is not possibly to totally disallow this type of data to be submitted to the application. A security application such as the OWASP Anti-Samy toolset should be implemented.

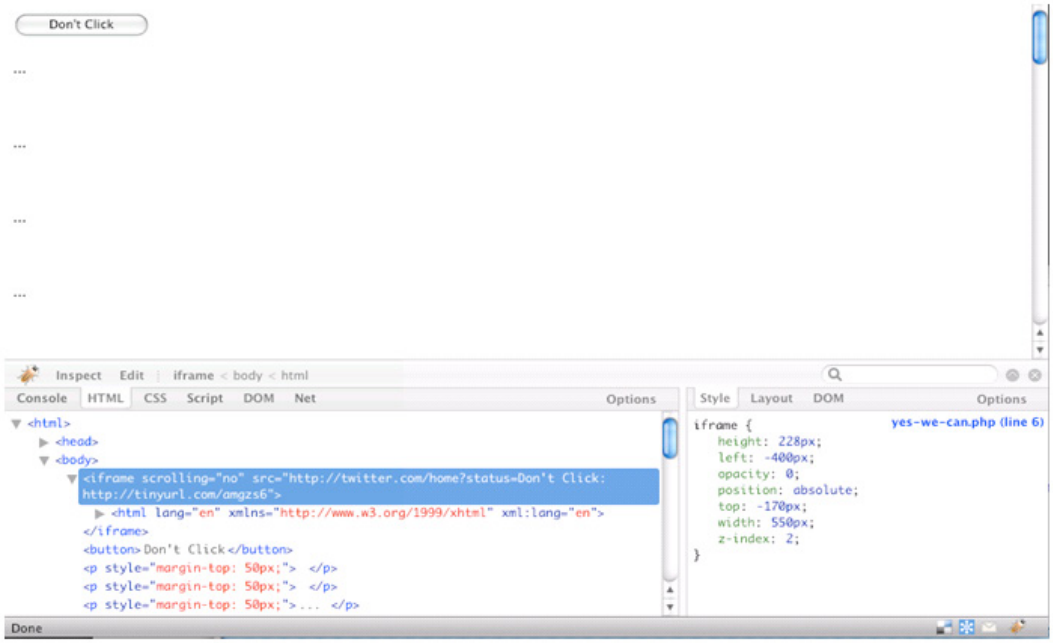
References

NBC Nightly News – Via social media, hackers weave a tangled web (http://www.msnbc.msn.com/id/3032619/#32467751)
DarkReading - Social networks number one web attack target (http://www.darkreading.com/security/attacks/showArticle.jhtml?articleID=219400520)
OWASP Anti-Samy (http://www.owasp.org/index.php/Category:OWASP_AntiSamy_Project)

TWITTER ATTACKS

Twitter was hit with a number of different attack methods throughout the year including: brute force authentication attacks, denial of service, click-jacking and CSRF/XSS worms. CSRF and XSS can be combined to create a strong blended attack, in this case a propagating worm. A Web 2.0 community generated site such as twitter is often vulnerable to stored XSS. This often implies that a user can update his own profile with malicious code and as a result others who view his content get hit. Without any other vulnerability to complicate things, you are safe as long as your friends are trustworthy.

However, if the site is also vulnerable to CSRF, the XSS exploit can include in addition to the payload also the original XSS inflicting code run under the attacked users credential, modifying his content and therefore hitting his own friends, which hit their own friends and so on.



Twitter may be a victim of its own success in that its popularity has caused cyber criminals to figure out methods of attacking its huge user base. Web 2.0 sites like Twitter are a "perfect storm" of functionality ripe for exploitation by attackers. Specifically, anyone can get an account for free which means that bad guys can interact with the application as a normal user in order to conduct analysis and identify vulnerabilities that may be exploited. Couple this level of access with the fact that it allows users to send dynamic data (such as javascript) and it is no wonder there have been a number of XSS types of attacks.

Lessons Learned

Organizations need to implement proper security to prevent these types of attacks as highlighted in the previous sections including insufficient anti-automation and sanitizing user-supplied data.

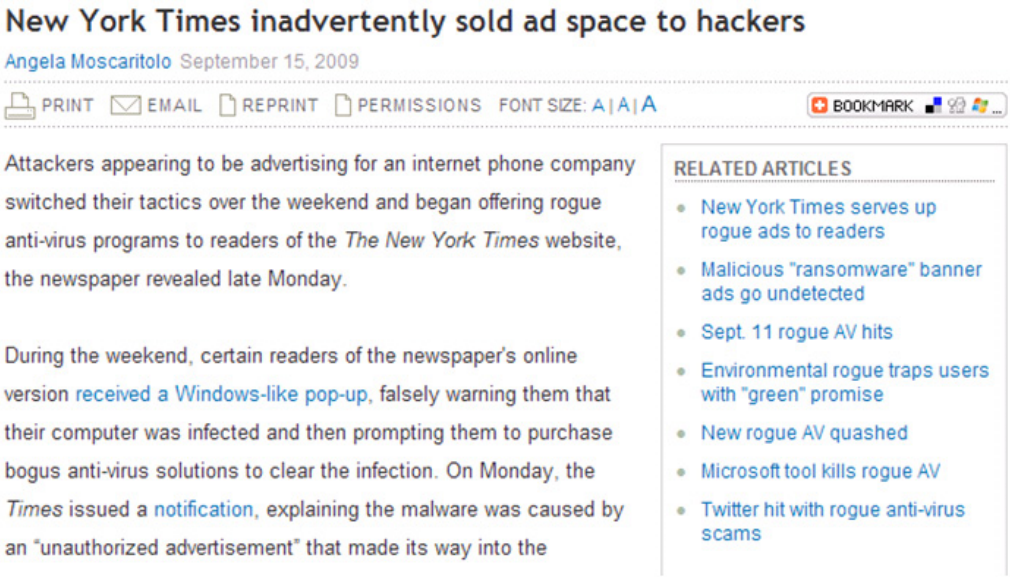
References

WHID 2009-2: Twitter accounts of the famous hacked (http://www.xiom.com/whid-2009-2)
WHID 2009-4: Twitter Personal Info CSRF (http://www.xiom.com/whid-2009-4)
WHID 2009-31: Double Clickjacking worm on Twitter (http://www.xiom.com/whid/2009/31/twitter_clickjacking)
WHID 2009-32: 750 Twitter Accounts Hacked (http://www.xiom.com/whid/2009/32/twitter_brute_force)
WHID 2009-37: Twitter XSS/CSRF worm series (http://www.xiom.com/whid/2009/37/twitter_csrf_xss)

SERVING MALWARE TO CLIENTS

Cyber criminals have expanded their attack methods to include compromising web sites, not to steal information, but rather to use the site as a malware depot. The goal is to use the site to try and infect their clients. The mass sql injection bots who injected malicious javascript into databases is a large percentage of the attacks, however other methods such as abusing banner ads has also increased.

Malware distribution through ad programs is a borderline phenomenon. While there is no question that malware distribution is malicious, and in most geographies illegal, in many cases the site owners are not technically responsible for the content of the ads they serve as the ad content comes directly from a 3rd party. The question whether they are legally responsible is open.



New York Times inadvertently sold ad space to hackers

Angela Moscaritolo September 15, 2009

PRINT EMAIL REPRINT PERMISSIONS FONT SIZE: A | A | A BOOKMARK

Attackers appearing to be advertising for an internet phone company switched their tactics over the weekend and began offering rogue anti-virus programs to readers of the *The New York Times* website, the newspaper revealed late Monday.

During the weekend, certain readers of the newspaper's online version received a Windows-like pop-up, falsely warning them that their computer was infected and then prompting them to purchase bogus anti-virus solutions to clear the infection. On Monday, the *Times* issued a notification, explaining the malware was caused by an "unauthorized advertisement" that made its way into the

RELATED ARTICLES

- New York Times serves up rogue ads to readers
- Malicious "ransomware" banner ads go undetected
- Sept. 11 rogue AV hits
- Environmental rogue traps users with "green" promise
- New rogue AV quashed
- Microsoft tool kills rogue AV
- Twitter hit with rogue anti-virus scams

Lessons Learned

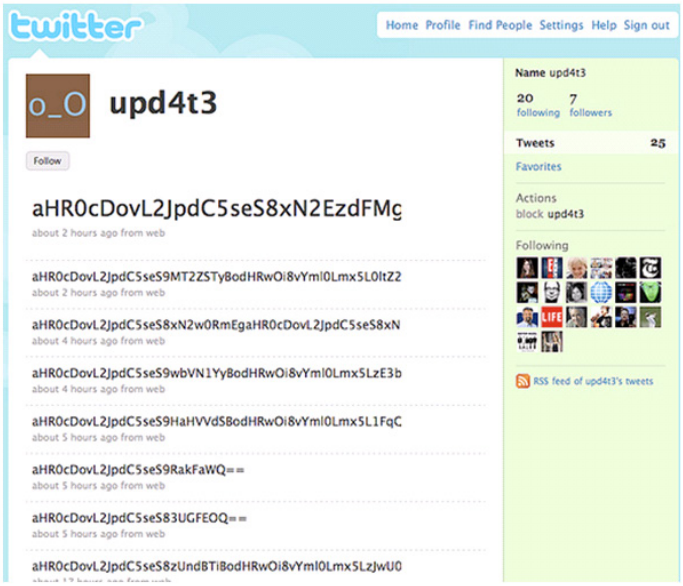
The underlying issue is web site reputation. If your clients feel that they cannot trust the data coming from your web site then they will go elsewhere. In order to address these issues, organizations have to implement some form of real-time, constant inspection of outbound data in order to identify if/when unauthorized modifications are made and it those changes are malicious.

References

WHID 2009-22: Federal Travel Booking Site Spreads Malware (http://www.xiom.com/whid/2009/22/federal_travel_booking_site_spreads_malware)
WHID 2009-12: Embassy of India in Spain found serving remote malware through IFrame attack (http://www.xiom.com/whid/2009/12/embassy-of-india-in-spain-found-serving-remote-malware-through-iframe)
WHID 2009-14: My.BarackObama.com Infects Visitors With Trojan (http://www.xiom.com/whid/2009/14/My.BarackObama.com_Infects_Visitors_With_Trojan)

BOTNET COMMAND AND CONTROL

Botnets are at the heart of most cyber criminal's arsenal and the bad guys are finding innovative methods to add legitimate servers to the zombie army. Servers provide more value to a botnet owner, as opposed to client systems, as they are always online and they normally have access to a much larger network pipe. While we have seen attacks such as Remote File Inclusion (RFI) used to draft a server into the botnet army, we are also seeing other innovative ways that attackers can abuse legitimate web site functionality in order to run a botnet command and control (C&C) server. Once such instance this year was when a botnet owner used a legitimate Twitter account for issuing botnet C&C information.



Botnets are at the heart of most cyber criminal's arsenal and the bad guys are finding innovative methods to add legitimate servers to the zombie army. Servers provide more value to a botnet owner, as opposed to client systems, as they are always online and they normally have access to a much larger network pipe. While we have seen attacks such as Remote File Inclusion (RFI) used to draft a server into the botnet army, we are also seeing other innovative ways that attackers can abuse legitimate web site functionality in order to run a botnet command and control (C&C) server. Once such instance this year was when a botnet owner used a legitimate Twitter account for issuing botnet C&C information.

Lessons Learned

Cyber criminals are actively looking to recruit web sites into their botnet armies. Not only should proper web application security protections be implemented, but network security ACLs should also be configured. For example, network firewall egress ACLs should be added to ensure that web servers are not able to initiate connections out to the Internet.

References

DarkReading – Botnet operators infecting servers, not just PCs (http://www.darkreading.com/vulnerability_management/security/app-security/showArticle.jhtml?articleID=222002433)
Wired – Hackers use Twitter to Control Botnet (http://www.wired.com/threatlevel/2009/08/botnet-tweets/)

PREDICTIONS FOR 2010

This section takes the information that we have seen thus far and helps us to look into the crystal ball at what lies ahead of us for 2010.

WEB-BASED WORMS MIGRATE OFF SOCIAL NETWORKING SITES

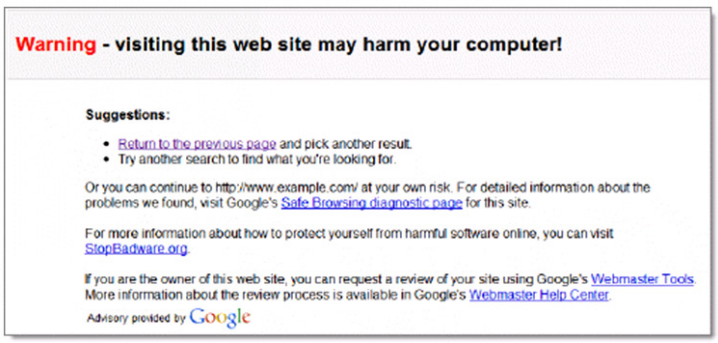
As we have seen from the previous sections, social networking types of web sites have fallen victim to web-based XSS/CSRF worms. It seems as though these types of web sites are a perfect testing ground for these types of attack mechanisms, however the attackers ideally want to migrate these attacks off to other types of web sites.

We believe that attackers will utilize Web 2.0 features such as RSS feeds, AJAX and widgets to propagate malicious code on other web sites. A Probable target for attackers, due to its enormous user base, is iPhone financial web apps such as:

- Mint
- Bank of America Online Banking
- E*Trade Mobile
- Bloomberg Mobile

PLANTING OF MALWARE BECOMES A TOP CONCERN

As mentioned in an earlier section, organizations can not afford to allow their web sites to serve malicious content to their customers. If this happens, consumer confidence will waiver and may cause them to move elsewhere. Another impact is that high profile web search engines such as Google may tag the web site as malicious and warn users. This negatively impacts Search Engine Optimization (SEO) efforts.



This is one of those scenarios that can directly impact the bottom-line such as stock prices. Due to this risk level – organizations will focus more efforts on security capabilities to inspect outbound content to ensure that it is non-malicious.

ATTACKS AGAINST WEB-BASED CRITICAL INFRASTRUCTURE COMPONENTS

It is no secret that terrorists and adversarial nation-states are seeking the capabilities to attack and disrupt critical infrastructures in the United States. Nuclear power plants, power grids, transportation control systems are all targets and they also share a similar capability – they often have web-based front-ends. The bad guys are seeking to exploit web-based flaws in order to be able to obtain access to data or the ability to shut down or cause a denial of service condition.

References

60 Minutes: Cyber War - Sabotaging the System (http://www.cbsnews.com/stories/2009/11/06/60minutes/main5555565.shtml)
SC Magazine - Web attacks can invade air traffic control systems (http://www.scmagazineus.com/report-web-app-hacks-can-invade-air-traffic-control-systems/article/136440/)

HTTP DENIAL OF SERVICE ATTACKS TAKE DOWN IMPORTANT SITES

Whereas network level DoS attacks aim to flood your pipe with lower-level OSI traffic (SYN packets, etc...), web application layer DoS attacks can often be achieved with much less traffic. Just take a look at Rsnake's Slowloris app if you want to see a perfect example of the fragility of web server availability. The point here is that the amount of traffic which can often cause an HTTP DoS condition is often much less than what a network level device would identify as anomalous and therefore would not report on it as they would with traditional network level botnet DDoS attacks.

Network DDoS attacks aimed at web sites can still be effective if the circumstances are right, however there are other web application specific types of attacks that are much more effective Network DDoS attacks aimed at web sites can still be effective if the circumstances are right, however there are other web application specific types of attacks that are much more effective while simultaneously requiring much less traffic. Odds are that there will be a number of high profile web sites that are knocked offline during 2010.

References

WHID 2009-1: Gaza conflict cyber war (http://www.xiom.com/content/whid-2009-1-gaza-conflict-cyber-war)
The Register – US websites buckle under sustained DDoS attack (http://www.theregister.co.uk/2009/07/08/federal_websites_ddosed/)
Cnet News – DDoS attacks hobble major sites, including Amazon (http://news.cnet.com/8301-30684_3-10421577-265.html?part=rss&subj=news&tag=2547-1_3-0-20)
Ha.ckers – Slowloris HTTP DoS (http://ha.ckers.org/slowloris/)

SUMMARY

Web application attacks are here to stay and the bad guys will continue to find innovative methods of fine tuning their attacks in order to try and achieve their goals. It is critical that organizations focus efforts on protecting their web applications from these types of attacks.

ABOUT THE AUTHOR

Ryan C. Barnett is Director of Application Security Research at Breach Security Inc., and leads Breach Security Labs. He is a recognized security thought leader and evangelist who frequently speaks with the media and industry groups. He is director of application security at Breach Security and a SANS Institute faculty member. He serves as the team lead for the Center for Internet Security Apache Benchmark Project and is a member of the Web Application Security Consortium. Mr. Barnett's web security book, "Preventing Web Attacks with Apache," was published by Addison/Wesley in 2006.

Ryan is available at Ryan.Barnett@Breach.com.

ABOUT BREACH SECURITY

Breach Security, Inc. is the leading provider of real-time, continuous web application security that protects sensitive web-based information. Breach Security's products protect web applications from hacking attacks and data leakage and ensure applications operate as intended. The company's products are trusted by thousands of organizations around the world, including leaders in finance, healthcare, ecommerce, travel and government.

For more information, please visit www.breach.com.