# Excuse Me But Is That a Gazebo On Your Site?!

**Lori MacVittie, 2009-01-10**

*There are few things in reality that can match The Gazebo in its ability to evoke fear and suspicion amongst gamers. The links on your web site may be one of them.*

In the history of Dungeons and Dragons there exists the urban legend known to all as "The Gazebo." The Gazebo, over the years, has become a gaming euphemism for a situation in which people over analyze and overestimate the risk involved with interacting with some "thing". In the case of The Gazebo the "thing" was, as you might guess, a gazebo. Yes, a simple wooden structure placed in gardens where lovers meet under the moon and all that. A player, according to legend, would not believe this simple gazebo was *not* dangerous. So he attacked it and, failing to elicit a response, eventually decides to run away. The DM (Dungeon Master), having been frustrated by the waste of time that was the encounter, decides the gazebo ***was*** a threat after all and has it eat the player's character.

Now it might be the case that gamers are just overly suspicious, as many types of geeks are wont to be. I will admit, with just a bit of embarrassment, that I was a part of a group of gamers who once frustrated Don for hours by treating some apparently innocent green algae as though it was a giant, poisonous snake. Yes, our "gazebo" was in fact just normal, everyday mold. Luckily for us Don was kinder than the DM in the Gazebo incident and we eventually realized how foolish we were and continued on with our game.

It may come as a surprise to you, but if you allow user-generated content on your site then thanks to circumstances beyond your control your users are probably running into Gazebos all over your site.

**THE GAZEBO ON YOUR SITE**

*with many apologies to my fellow gamers and especially Richard Aronson*

**Web Master:** You see a well-designed web site. In the middle, on a post, you see a link.

**Eric:** A link? What color is it?

**Web Master:** (Pause) It's blue [default 'unvisited link' color], Eric.

**Eric:** How far away is it?

**Web Master:** About half way down the page.

**Eric:** What's the domain name?

**Web Master:** (Pause) It's thislinkisokaytoclickonipromise.com.

**Eric:** (clicks mouse) I view source to detect whether it's good.

**Web Master:** It's not good or bad, Eric. It's a link!

**Eric:** (Unusually long pause, even for Eric) I put my mouse over it.

The latest study "State of Internet Security" from WebSense indicates that 95% of all user-generated content is, well, to put it simply, "bad". Even more frightening is the conclusion that "61 percent of the top 100 sites either hosted malicious content or contained a masked redirect" and "77 percent of Web sites with malicious code are legitimate sites that have been compromised."

Basically, the Internet is full of Gazebos and it's enough to make users shy away from clicking on *any* link on *any* site lest they become infected with the latest malware du jour.

The InfoSec community spends a lot of time talking about how businesses can protect themselves against miscreants, but we don't often talk about how we can protect our users from, well, other users. Yet according to the WebSense study and "top ten lists" of attack techniques, it is user-generated content that puts both business and its users at risk for malware, for attack, for theft of identity and personal information. That's probably because we can control many of the variables that put the business at risk but there's less we can do to protect users from other users and themselves.

## THERE'S NO FOOL-PROOF SOLUTION TO THIS ONE

The use of user-generated content as a means to exploit vulnerabilities in both client and server side systems means that the first line of defense should be at the web-application, at the point at which the user is generating the content. Simply disabling the ability to share information via links is not an option today as the majority of sites are based entirely on the is capability and without links the

**Web Master:** It says “Follow me”. It's a link!

**Eric:** (Pause) I close the source view and open my anti-virus scanner. Does it respond in any way?

**Web Master:** No, Eric. It's a link!

**Eric:** I run the anti-virus scanner. What happened?

**Web Master:** You are now using 80% of your CPU to run anti-virus.

**Eric:** (Pause) Didn’t it neutralize it?

**Web Master:** Of course not, Eric! It's a link!

**Eric:** (Whimper) But the anti-virus should detect if it’s malicious or not!

**Web Master:** It's a link, Eric, a link!

**Eric:** (Long pause - he has no more ideas) I close the page.

**Web Master:** (Thoroughly frustrated) It's too late. You've awakened the link, and it automatically downloads a virus that eats all the data on your hard drive.

**Eric:** (Reaching for his CDs) Maybe I'll install Linux so I can avenge my Windows install...

Internet essentially breaks.

Now if the link being submitted or included in the user-generated content contains something “evil” it’s easy enough for a web application firewall (WAF) or the application’s own security checks to stop it from being added to the system and later propagated out to users. A WAF can determine when someone is trying to inject a malicious link into a site via XSS or SQLi or through obfuscation and stop ***that*** from happening, but if the link is “just a link”, there’s really no good way to determine its “goodness” or “badness” without following it and examining ***its*** content and environment.

But links are neither “good” nor “bad” themselves, they’re just a mechanism for connecting (integrating) two disparate sites together. It’s the content ***behind*** the links that’s the problem, and that’s something that’s far more difficult to ascertain when the content is somewhere else. If it’s just a link and someone is trying to entice a user to visit it and it is at the destination site where “bad” content resides, neither a WAF nor the application’s security checks can really address the problem.

We’ve solved this problem, to a large degree, with e-mail and SPAM already through the use of reputation-based systems. These systems evaluate the reputation of the sender and, based on that information, determine whether the mail will be accepted or not. Now we can’t necessarily do that with users generating content but we could do something similar to that with links. If you’ve ever read through descriptions of worms and viruses and links that spread malware you’ll note that the common theme across all the links is that they’re going to one of a short list of URLs with some identifying characteristics.

It is those identifying characteristics we could use to determine the “goodness” or “badness” of the link and thus either allow or deny the user to include it in their user-generated content. If we already know there is a scam going around we can use network-side scripting to update a list of URLs or those identifying characteristics so that as the content is being generated we can scan the content for those URLs and if we find one of the “bad” ones, refuse to add the content to our site. But that assumes we ***know*** what the “bad” URLs and domains already are, which is not always the case. If we don’t already recognize a domain as “bad”, we really can’t do much about it. We have to assume it’s good and let it pass.

But if we take the concept of metadata hubs sharing information across the Internet we could easily apply this to sharing “bad link” information and thus eliminate the manual processes that require solutions be updated by hand every time a new “bad link” is discovered. David O’Berry first suggested this concept as a means to create a threat distribution channel for InfoSec and that idea is applicable over a wide variety of “threats” – including “bad links”. A more real-time approach to sharing information regarding “bad” domains might improve the situation, but it remains that applications and security infrastructure would need to take advantage of that data and that’s a capability no one really has today.

So basically no single solution has the answer to this one. It’s going to require a combination of solutions – some of which do not exist today – to reduce the risk of shared, user-generated content. The only thing that is certain is that we need to address the problem before users become so paranoid that they refuse to click on ***any*** link. Because that, my friends, would be the end of the game, er Internet.

- Dungeons and Dragons Gaming Stories Including Eric and the Gazebo
- AJAX and Network-Side Scripting
- Understanding ***network-side*** scripting
- If Your Users See an HTTP Error Code You're Doing It Wrong
- I Can Has UR .htaccess File
- Automatically Removing Cookies
- Clickjacking Protection Using X-FRAME-OPTIONS Available for Firefox
- Stop brute force listing of HTTP OPTIONS with network-side scripting
- Jedi Mind Tricks: HTTP Request Smuggling
- I am in your HTTP headers, attacking your application

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com