The Problem with Web Links in Email

Why are web links in email messages a problem - and what can be done to combat it

DEEP SECURE

What's the problem with the web links in email? You click them, they take you somewhere dangerous and you are in trouble. Who is to blame? Well the user did the clicking and surely, they should know better. We really should train them to not click on dangerous links. But links are made to be clicked. They are useful. We want to click them. We need to click them. Except when they are dangerous...

Simply Fix It

The problem is the user is in no position to tell a safe link from an unsafe link, because the bad guys are so good at disguising an attack. Training users to not click on links is like training them to not do their work. So, if we can't fix the problem by reprogramming the users, should we be expecting the mail gateway team to do something more? After all, they are the ones who let the link in in the first place. Surely all they have to do is find the links, check whether they are safe and block them if not. Yes, but easier said than done.

Links are not always in obvious places – they can be embedded in documents attached to the email, they can be in calendar requests, they can even be bits of ordinary text that the receiving application interprets as a link to help the user.

Even if you find the links, you then have to decide if a click would be dangerous. If the link refers to a known malware site or a known fake website that's phishing for passwords then it is easy – find the link on a list of bad sites and block it.

The problem comes when the attacker is a little bit smarter. What if the malware is hosted on a site that is legitimately used to share safe content needed by the user – blocking the link because it might be unsafe results in blocking essential safe content.

link because it might be unsafe results in blocking essential safe content.

Perhaps you try harder and actually follow the link to download the referenced content and check that – well that's ok if the content is a document, like a PDF, but not if it's a web page.

A web page can contain scripts that can generate the unsafe content when run and it's easy for the script to look harmless when being checked by the mail gateway, only dropping the malware when run on the user's desktop. It's also possible for the attacker's website to notice that a mail gateway is fetching the content, rather than the user's browser, and actually return safe content. Only when fetched by the browser is the unsafe content returned.

Solution: Re-Writing Web Links

One way out of this is to rewrite the web links so they refer to a special security gateway that checks the links as they are clicked. The user's click now takes them to the gateway, passing the original link as a parameter. The gateway looks at the parameter and decides if it is safe. If so, the browser is redirected to the site, and if not, an error page is returned.

This works, assuming (1) it is possible to find all the links to rewrite them, that (2) an attacker's website cannot tell the difference between the web gateway checking and the browser fetching, and (3) the resulting content is obviously unsafe. In practice though, this is all a bit hit and miss.

Regards stopping malware, rather than phishing for passwords, having a mail gateway check web links is weak because the malware lives in the content that is actually retrieved, and it's impossible to predict what is going to arrive by looking at a mail message. We need something more.

Solution: Web Gateways

If there's a web gateway in the mix then we don't need to worry what's in the email because any unsafe content is going to pass through the web gateway and that can take care of it.

In the past, the web gateway was not in a strong position to block malware, because web browsing involved downloading and executing mobile code that was run by complex applications. Flash, ActiveX and scripted documents like PDFs were essential to bringing web content alive, but they gave attackers an easy route into a system. But that's a thing of the past, largely due to HTML5 and CSS3 – browser technologies that deliver active web content in a safe way. A web gateway can now block other active content without breaking the web. It doesn't need to know if that active content is unsafe - it is always unnecessary and potentially dangerous, so it gets blocked.

But malware can also be found in passive data – documents, images, etc. This generally exploits flaws in applications that cause specially crafted malformed data to be executed. The web gateway needs to block such malware as well. This can be done by examining the data to decide if it is unsafe, but that fails to detect a new form of malware, or in some cases even a trivial modification of known malware.

Rather than trying to spot malware, the alternative is to assume all data in unsafe and build new data known to be safe to be delivered in its place. That way an attacker's data is never delivered, so they have no way of fooling the gateway into letting it pass. A web gateway that does this protects against malware regardless of what links the user finds and clicks.

Protecting Passwords

The web gateway does not solve the issue of phishing websites that are looking to steal the user's password, other than to block access to known phishing sites. But that's really a problem with having users enter passwords to authenticate themselves to web sites. Again, modern browsers and web apps do a lot more to defend against this than a mail/web gateway could ever hope to achieve. Having the browser remember passwords for particular sites means the user doesn't have the opportunity to type them into a fake site, and two factor authentication (done properly) reduces the significance of the passwords.

Summary

Users should be able to click links in email without endangering the system. It's the web gateway that provides the main defence, not the mail gateway. It blocks malware, preferably by building safe content rather than trying to spot unsafe content and stops access to obvious phishing sites. And it's the browser with a modern password strategy that saves users from the smarter phishing sites. The mail gateway still needs to stop malware in attachments, again its best if it does this by building safe content, but it doesn't need to worry about links.