ars

SIGN IN

RISK ASSESSMENT —

New attack bypasses HTTPS protection on Macs, Windows, and Linux

Hack can be carried out by operators of Wi-Fi hotspots, where HTTPs is needed most.

DAN GOODIN - 7/26/2016, 10:14 AM

Pdac

67

A key guarantee provided by HTTPS encryption is that the addresses of visited websites aren't visible to attackers who may be monitoring an end user's network traffic. Now, researchers have devised an attack that breaks this protection.

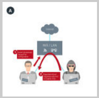
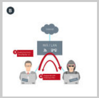
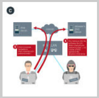
The attack can be carried out by operators of just about any type of network, including public Wi-Fi networks, which arguably are the places where Web surfers need HTTPS the most. It works by abusing a feature known as WPAD—short for Web Proxy Autodisovery—in a way that exposes certain browser requests to attacker-controlled code. The attacker then gets to see the entire URL of every site the target visits. The exploit works against virtually all browsers and operating systems. It will be demonstrated for the first time at next week's Black Hat security conference in Las Vegas in a talk titled Crippling HTTPS with Unholy PAC.

"People rely on HTTPS to secure their communication even when the LAN/Wi-Fi cannot be trusted (think public Wi-Fi/hotels/cafes/airports /restaurants, or compromised LAN in an organization)," Itzik Kotler, cofounder and CTO of security firm SafeBreach and one of the scheduled speakers, wrote in an e-mail. "We show that HTTPS cannot provide security when WPAD is enabled. Therefore, a lot of people are actually exposed to this attack when they engage in browsing via non-trusted networks."

With the exception of the full URL, all other HTTPs traffic remains unaffected by the attack. Still, in some cases, disclosure of the URL can prove fatal for security. The OpenID standard, for instance, uses URLs to authenticate users to the sites and services that support it. Another example is document sharing services, such as those offered by Google and Dropbox, that work by sending a user a security token that's included in the URL. Many password-reset mechanisms similarly rely on URL-based security tokens. Attackers who obtain such URLs in any of these cases are often able to gain full access to a target's account or data.

The most likely way the attack might be carried out is for a network operator to send a malicious response when a computer uses the dynamic host configuration protocol to connect to a network. Besides issuing addresses, DHCP can be used to help set up a proxy server that browsers will use when trying to access certain URLs. This attack technique works by forcing the browser to obtain a proxy autoconfig (PAC) file, which specifies the types of URLs that should

A second way to execute the attack is through use of malware that modifies the targeted machine's network settings to use a proxy. In either case, people targeted by the attack will see few overt signs anything is amiss, and the browser address bar will continue to show the HTTPS connection is valid. The notable exception is that, under the first scenario, a computer's networking options (for instance, a Mac's Proxies tab in the advanced part of the system preferences network settings) will show it's using proxy auto-discovery with no further details. In the event the attack was carried out using malware, the computer's network configuration will show it's using proxy auto-configuration and will also display the attacker's URL.

It's by no means the only attack to abuse WPAD and PAC. In April, researchers documented how a banking Trojan known as BlackMoon installs a PAC file on infected computers to redirect browsers to phishing pages when accessing targeted sites. The file also contains a JavaScript function that BlackMoon abuses to intercept user credentials and to obfuscate the specific banking sites that are being targeted. A separate talk scheduled at next week's Black Hat is titled BADWPAD.

"The speaker will share how his team initially deployed a WPAD experiment to test whether WPAD was still problematic or had been fixed by most software and OS vendors," the Black Hat site stated. "This experiment included attacks in 1) Intranets and open-access networks (eg Free-WIFI spots and corporate networks) and 2) DNS attacks on clients leaking HTTP requests to the Internet."

The vulnerability that the Unholy PAC attack exploits resides in the WPAD specification, which was first drafted in 1999. That means it's not easily fixed by either OS or browser makers. Still, browsers can largely work around the vulnerability by following the lead of Microsoft's Edge and Internet Explorer 11 browsers, which invoke the FindProxyForUrl function with URLs that are truncated to host names only, as opposed to full URLs, which may contain authentication tokens or credentials.

"By doing so, the browser eliminates the exposure of protocol data to the minimum (host names only), which is already exposed to the world via DNS queries and as such is not considered too sensitive," Amit Klein, vice president of security research at SafeBreach, said in an e-mail. End users can also take steps to disable WPAD, but that remedy may not work in situations where it's required to connect to a given network.

Post updated to change "cripples" to "bypasses" and "crypto" to "protection" in the headline.

Promoted Comments

InfoSecGuy / Smack-Fu Master, in training

JUMP TO POST

> nibb wrote:
>
> > InfoSecGuy wrote:
> >
> > > mmiller7 wrote:
> > >
> > > Is it that big of a deal to have the URL discovered?

> > > payment information is secure.
> >
> > It's not uncommon for web forms to include usernames and even passwords in the URL. Especially since a lot of web devs have grown lazy because HTTPS was supposed to protect that information.
>
> In 1997 maybe. Today, anyone doing that is logging the passwords in the web server logs as well. In plain text !

I've seen three sites this week doing it. Additionally I've seen several of my companies internal apps doing it. It's quite common today.

29 posts | registered 6/1/2015

---

guest435346 / Smack-Fu Master, in training **JUMP TO POST**

Many federated authentication protocols send security tokens from one domain with a redirect through the user's browser to another domain. In this regard, OAuth 1, OAuth 2 (and thus OpenID Connect) and SAML2 are basically all variations on the same theme: in the end the browser performs a GET request with some kind of security token in the URL.

Incidentally, there's no problem with logging this token, because it should be good for one use only.

I don't think it's a vulnerability in all of these specifications. Or at least, it's a bit easy to sit here in your armchair and call all these specifications stupid and vulnerable. I think Microsoft's fix is actually pretty sensible. Good job Microsoft!

2 posts | registered 7/4/2016

---

lewax00 / Ars Scholae Palatinae / *et Subscriptor* **JUMP TO POST**

> beebee wrote:
> Maybe I'm over thinking this or don't understand https, but DNS is needed to convert the domain to IP, so who you are communicating with isn't a secret.

The domain yes (e.g. "arstechnica.com"), but DNS does not see the full url (e.g. the "/security/2016/07/new-attack-that-cripples-https-crypto-works-on-macs-windows-and-linux/" part is only seen by the server at "arstechnica.com" via HTTPS). You don't mentally separate them because they're generally seen as one string, but in the relevant protocols those pieces are separate.

6740 posts | registered 6/20/2013

DAN GOODIN
Dan is the Security Editor at Ars Technica, which he joined in 2012 after working for The Register, the Associated Press, Bloomberg News, and other publications.

EMAIL dan.goodin@arstechnica.com // **TWITTER** @dangoodin001

READER COMMENTS 67 SHARE THIS STORY

← PREVIOUS STORY NEXT STORY →

Related Stories

ars SIGN IN

Today on Ars

RSS FEEDS
VIEW MOBILE SITE
VISIT ARS TECHNICA UK
ABOUT US

CONTACT US
STAFF
ADVERTISE WITH US
REPRINTS