

Without a **trace**

Foiling application fingerprinting using the one-two punch of BIG-IP and iRules.

APPLICATION SERVER IDENTITIES have always been a relatively easy way to deduce what is actually serving up your application content. While this is a nice feature for most functions of application data transactions, it is also a very desirable feature for targeted malicious attacks. Attacks always start by looking for the lowest-hanging fruit first; known targets are easier to attack than unknown targets. If an exploit exists against ACME's WebServer 4.021, for example, you need only query thousands of public Web servers until you find one that responds with its unique identifier ("Server: ACME WebServer 4.021"), and you're in.

To move beyond server-string reliance and evaluation, application fingerprinting tools have become a topic of interest within information security circles. Application fingerprinting is based on the logic that applications of the

is running, regardless of what string values for "Server:" it reports.

Enter BIG-IP Local Traffic Manager v9 and iRules. With this powerful combo, you can break the cycle of repetitive response patterns from appli-

Beyond server-string reliance, application fingerprinting tools have become a topic of interest within information security circles.

same family typically respond in the same way with measurable repeatability. Knowing how an application server is going to respond lets application fingerprinting tools more reliably deduce what server

cation servers without breaking the application or touching the back-end servers. The iRule available on DevCentral is an example of a few simple ways to control how protocol information is returned to the

user and how to obscure true application server identities by attempting to confuse programmatic fingerprinting tools in a number of ways.

Confusing any fingerprinting tool is not always easy, and isn't always desirable from a

production standpoint. Some applications require a valid Server: header string in or-

der to work properly. However, if you don't want to notify the world that you're running Apache 2.0/PHP 4.3.9 or IIS/6.0 with .NET 2.0.5, and you wish to hide this information from automated attack

reconnaissance tools, this small rule shows how you can make a huge impact in keeping your websites secure by further obscuring your server's true identity.

The new iRule* is free and available now. For more information, visit DevCentral and the Obscure HTTP Server Identities iRule.

***NOTE: Although it has been tested against multiple servers, such as versions of Apache and IIS, and multiple applications, such as OWA, SharePoint, and PHP, with extremely high rates of success in fooling application fingerprinting tools, this rule should be used only as an example.**



DCTV and iTunes

WHEN WE ROLLED out DevCentral Television (DCTV), we honestly had no idea how popular it would be. More than fifteen months and tens of thousands of views later, we're expanding the topics we cover with DCTV. We're also expanding the ways we reach viewers. Because so many people have the ever-popular iPod, we've decided to syndicate our DCTV videos—which now include security experts and topics as well as the usual suspects—through iTunes. Now, you can download the newest DCTV to your video iPod and get the latest dose of technical TV wherever your travels take you. You'll laugh, you'll cry ... and maybe even learn a new iRule command or two.

About F5's iRules and iControl API

F5's iRules represent a revolutionary approach to intercepting, parsing, modifying, and routing application traffic based upon extremely customizable logical policies. iRules, exclusive to F5 Networks BIG-IP v9, utilize a standard scripting language—Tool Command Language (TCL)—to construct these policies that you can apply to any

IP application. You can use many of the standard TCL commands—plus a robust set of extensions that BIG-IP provides to help administrators and application developers further optimize and secure application traffic—to meet specific requirements, expand upon an application's functionality, or even offload some of the work to the network layer. For more information, visit F5's online developer community, DevCentral, at: <http://devcentral.f5.com>.

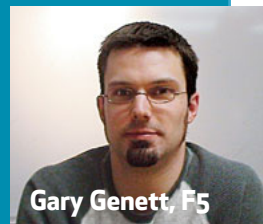


NEW ONLINE VIDEO:

Login to your FirePass using simple Unix tools, complete with script.

It only seems like magic.

FirePass doesn't use any proprietary voodoo to generate SSL VPN tunnels for *Nix systems and their users. The underlying technology is nothing more complex than PPP over SSL, coupled with unique session-based authentication.



Gary Genett, F5

Internally, F5 has used this to our advantage for testing purposes and employee remote access from Unix-style systems not officially supported by FirePass. Over the years, we have developed several very robust and complex testing suites and end-user access tools using OpenSSL and PPPD. We took the foundation of our internal tools and turned it into a publicly available Perl script. This fully functional example performs the three steps necessary to login to FirePass and create an SSL VPN tunnel.

Learn more about this process, including where you can get your hands on some fully functional source code, in this DCTV interview.

Learn how in a
short DCTV
video from F5:
<http://devcentral.f5.com>